IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 24, 2022, accepted May 15, 2022, date of publication May 18, 2022, date of current version May 25, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3176106*

# A Survey of State-of-the-art on Edge Computing: Theoretical Models, Technologies, Directions, and Development Paths

**BIN LIU[1], ZHONGQIANG LUO[1,2], (Member, IEEE), HONGBO CHEN[3], AND CHENGJIE LI[4]**

[1]School of Automation and Information Engineering, Sichuan University of Science and Engineering, Yibin 644000, China
[2]Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Yibin 644000, China
[3]Sichuan Shuneng Electric Power Company Ltd., Chengdu 610000, China
[4]School of Computer Science and Technology, Southwest Minzu University, Chengdu 610041, China

Corresponding authors: Zhongqiang Luo (zhongqiangluo@gmail.com) and Hongbo Chen (chb740921@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61801319, in part by the Sichuan Science and Technology Program under Grant 2020JDJQ0061 and Grant 2021YFG0099, in part by the Sichuan University of Science and Engineering Talent Introduction Project under Grant 2020RC33, in part by the Innovation Fund of Chinese Universities under Grant 2020HYA04001, in part by the Artificial Intelligence Key Laboratory of Sichuan Province Project under Grant 2021RZJ03 and Grant 2021RZJ04, and in part by the 2021 Graduate Innovation Fund of Sichuan University of Science and Engineering under Grant y2021071.

**ABSTRACT** In order to describe the roadmap of current edge computing research activities, we first address a brief overview of the most advanced edge computing surveys published in the last six years. It is true that edge computing has been adaptively integrated into growing number of applications. Edge computing theory and technology will bring substantial innovation and incentive, as well as a large number of application scenarios in different fields, such as edge computing assisted smart city, Internet of Vehicles(IoV), Industrial Internet, and many other different fields. In the field of edge computing, however, it is actually lack of a comprehensive investigation of using the most advanced theoretical models, technologies, directions and development paths. To fill this gap, by identifying and classifying, we carry out an in-depth survey of the latest high-quality literatures related to the theoretical discoveries in edge computing(EC) and the fusion of EC and the frontiers of Information and Communication Technology (ICT). Finally, it is summarized several promising open issues, and also pointed forwards the directions of future research. We hope that this survey report will attract much more attention, stimulate fruitful discussions, and provide ideas and useful guidance for further research on the theoretical models, technologies, directions and development paths of edge computing.

**INDEX TERMS** Blockchain, computation migration, edge computing, edge intelligence, resource scheduling.

## I. INTRODUCTION

In recent years, as the continuous increase of data volume and the diversified requirements of data processing, cloud-based big data processing has faced many challenges. The Internet of Things(IoT) era has brought higher requirements for transmission bandwidth, latency, energy consumption, application performance and reliability. In this context, it is difficult to meet the high-performance requirements of users due to the limited bandwidth, high latency and high energy consumption of the centralized processing model of cloud computing.

The associate editor coordinating the review of this manuscript and approving it for publication was Qichun Zhang.

Fortunately, Information Technology (IT) resources can be migrated from the traditional cloud data center to the user side, shortening the physical distance between users and information technology resources, achieving lower data interaction latency and saving network traffic, so as to provide users with low latency and high stability IT solutions. With the edge features, edge computing has penetrated into multiple applications that are closely related to all aspects of our daily life, such as autonomous drive, smart cities, industrial areas, and commercial applications. In the following, before discussing the motivation of this survey, we first provide a brief description of the latest edge computing survey articles published in recent years.

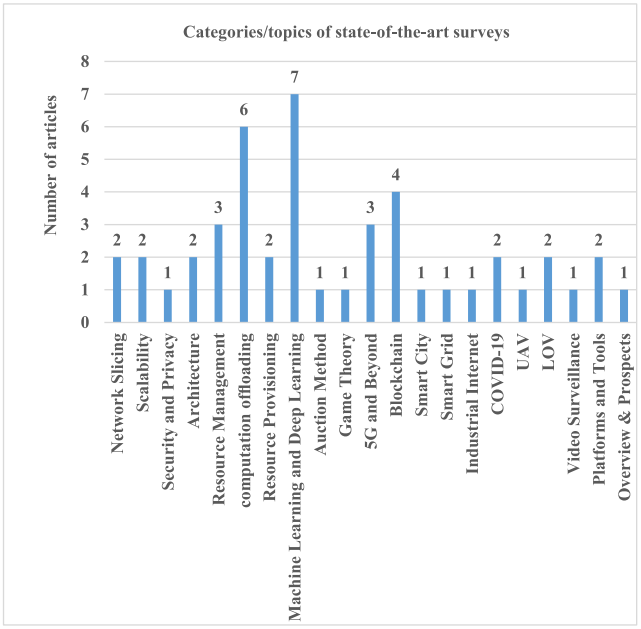


**FIGURE 1.** The latest categories and numbers of edge computing-related surveys published in recent years.

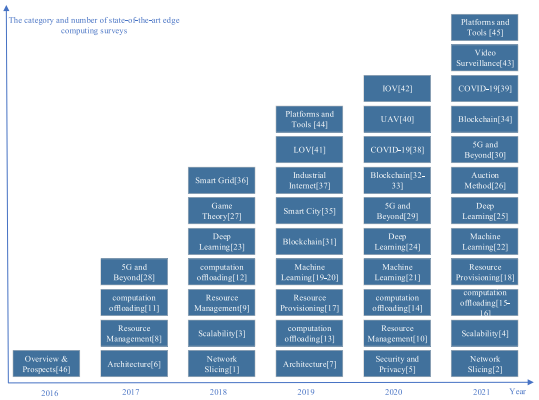


**FIGURE 2.** The latest categories and numbers of edge computing-related surveys organized in a chronological order.

### *A. TAXONOMY OF STATE-OF-THE-ART EDGE COMPUTING SURVEYS*

To clarify our survey objectives, we collect 46 representative survey articles related to edge computing. Fig. 1 shows the number of surveys in each of these categories. We learn that the most popular topics are computation offloading, machine learning(ML) and deep learning(DL), and blockchain. We also classify it in Fig 2. From the figure, we find that the number of published surveys increases over time, and the diversity of topics is growing. Specifically, we summarize the groups, categories, years and corresponding topics of these surveys in Table 1 and Table 2. For this, we divide these surveys into the following seven groups. The general principles of collection are based on the dissimilar aspects of edge computing covered in the survey. In group 1, we focus on the basic properties of edge computing. In group 2, by analyzing the resource scheduling work in edge computing, which is divided into resource management, computation offloading and resource provisioning. In group 3, the deployment of edge computing often faces the characteristics of complex environment and scattered sites. Different decision-making technologies can be adopted to make reasonable and efficient choices. Group 4 is the integration of edge computing and the latest cutting-edge communication technologies. In groups 5 and 6, we review the application scenarios of edge computing and related open-source tools and simulation platforms. Group 7 is the overall overview of the work.

#### 1) THE PROPERTIES OF EDGE COMPUTING

The first group is related to edge computing properties. About network slicing, scalability, security and privacy, and architecture are reviewed and summarized in [1]–[7]. For example, End-to-End (E2E) network slicing facilitates resource allocation for a wide variety of applications, and Rajkumar *et al.* [1] extended this to fog and cloud computing by examining state-of-the-art slicing approaches in network technologies such as SDN and NFV. Shah *et al.* [2] provided MEC and network slicing use cases for 5G services compared to [1], detailing the latest advances, implementation techniques, solutions and standardization efforts to achieve E2E network slicing. Pan *et al.* [3] envisioned a secure and scalable Fog-IoT architecture, such as the development of new automation tools with appropriate abstractions to allow the deployment of IoT applications on decentralized Fog nodes. Kumar *et al.* [4] discussed multiple algorithms, objectives, approaches and their benefits used in edge/fog computing due to limitations such as bandwidth limitations, inactivity, lack of resources and various security issues, laying the foundation for system scalability and security improvement. Alwarafy *et al.* [5] summarized the security and privacy of edge computing in IoT, which includes the key classification, types and countermeasures for attacks. Abbas *et al.* [6] highlighted the definition, advantages, architecture and development of related technologies of MEC. Hamm *et al.* [7] characterized current edge computing initiatives, giving a roadmap for sustainable edge computing.

#### 2) RESOURCE SCHEDULING

In MEC systems, resource scheduling has been a important issues in academia as they relate to saving energy and reducing latency of the entire edge computing system. The existing survey studies are shown as follows. Luong *et al.* [8] provided an overview of resource management in cloud networks applying economic and pricing models and shared resources in edge computing using pricing strategies. Recently, Hong *et al.* [9] have conducted a comprehensive survey and discussed some future research directions for improving resource management aspects to address the remaining challenges. Ghobaei *et al.* [10] presented a

systematic literature review of resource management methods in fog computing in the form of a classical taxonomy divided into six main categories. Mach *et al.* [11] divided the computation offloading domain in edge computing into three parts: decision making, resource allocation, and mobility management. Wang *et al.* [12] employed a new feature model to investigate the key problem of offloading in the edge-cloud framework, and listed the related methods and efforts. Dziyauddin *et al.* [13] conducted a comprehensive survey of computation offloading and data caching and delivery methods, as well as a detailed review, discovery and comparison of their existing optimization techniques. Wang *et al.* [14] conducted a detailed survey of 71 published studies related to task offloading in edge-cloud computing, classifying their tasks in the field of type, offloading scheme, target, portability and respectively. Algarni *et al.* [15] compared different computation offloading models in order to optimize the unloading parameters and techniques. Furthermore, several commonly used unloading algorithms were compared in terms of cost, time, and energy. Islam *et al.* [16] divided different task offloading models into three main categories and reviewed the recent research on MEC task offloading. Duc *et al.* [17] summarized the approach to ML techniques for solving the problem of reliable resource provisioning in edge-cloud environments. Spinelli *et al.* [18] discussed the flexibility of MEC deployments, migration capabilities and several use cases in industrial verticals.

#### 3) DECISION-MAKING TECHNOLOGIES

Edge computing can bring many advantages in combination with other domains, such as edge computing combined with machine learning and deep learning [19]–[25], auction methods [26], and game theory [27]. For example, to address the bottleneck problem of edge learning, Wen *et al.* [19] focused on a new class of data-importance aware RRM techniques and its recent advances in the field of edge intelligence(EI). Murshed *et al.* [20] concluded a review of edge-based techniques for training and deployment of ML. Shakarami *et al.* [21] classified various principles of ML-based offloading mechanisms for reinforcement learning, supervised learning and unsupervised learning. Shuja *et al.* [22] comprehensively studied ML-based edge caching techniques on the basis of [21]. Voghoei *et al.* [23] mapped DL to an edge computing paradigm with some related discussions. Wang *et al.* [24] focused on the scenarios and basic enabling techniques for DL and MEC applicability. Sun *et al.* [25] provided a comprehensive survey of DDL in terms of decentralization techniques, communication efficiency and trustworthiness, providing knowledge of privacy protection in multiple massive amounts of various types of edge computing data. Qiu *et al.* [26] identified an overview of auction methods in edge computing and a detailed review, analysis and comparison of their differences. Moura *et al.* [27] mainly reviewed the challenges and usage scenarios of applying game theory to MEC services.

#### 4) NEW COMMUNICATIONS NETWORKING

As the commercialization of 5G continues to accelerate, MEC, a key part of the 5G architecture, has received a lot of attention. Taleb *et al.* [28] focused on the basic key technologies to implement MEC, summarizing the 5G standards supporting MEC and IoT applications. Pham *et al.* [29] covered the basic principles of MEC and the latest research on its integration with 5G and other technologies. Al-Ansi *et al.* [30] thoroughly investigated the characteristics, advantages, challenges and potential use cases and market drivers of Intelligent Edge Computing (IEC). Its integration into 5G and 6G technologies can provide adequate and large-scale support for different services. Yang *et al.* [31] introduced an integrated system for blockchain and edge computing in terms of their framework, network security, data integrity and computational validation. Nguyen *et al.* [32] focused mainly on a broad discussion of the potential of blockchain to enable key technologies such as 5G, edge computing, SDN, NFV and network slicing, etc. Queiroz *et al.* [33] applied blockchain and edge computing to current IoV solutions and classified different scenarios, designs, algorithms and technologies. Gadekallu *et al.* [34] explored the developments and security challenges of blockchain and edge of things (BEoT) technology and analyzed its application in industrial applications.

#### 5) EDGE COMPUTING APPLICATIONS

Edge computing has spawned a large number of applications in various fields. The existing surveys on edge computing-based applications cover research areas such as smart cities [35], smart grids [36], Industrial Internet [37], Corona Virus Disease 2019 (COVID-19) [38], Unmanned Aerial Vehicles (UAV) [40], Internet of Vehicles [41], video surveillance [43], etc. The existing surveys are as follows. Khan *et al.* [35] provided a comprehensive study of edge computing in smart cities. Boccadoro *et al.* [36] investigated the application of fog computing in smart grids in terms of features, solutions, and challenges, providing a powerful tool for designing optimized smart grid systems. The key technologies of MEC for the Industrial Internet have been outlined by Li *et al.* [37], whose feasibility and importance were demonstrated by typical industrial applications that have been deployed. Sufian *et al.* [38] explored the significant impact of COVID-19 on global health, economy and education, and recalled that the use of technologies such as deep migration learning and edge computing to automate infrastructure is helpful to deal with the epidemic. Bianco *et al.* [39] investigated the close combination of RFID system and EI paradigm to address the outbreak of future pandemics. Zhou *et al.* [40] provided a comprehensive survey of recent advances in the field of MEC networks for UAV. Zhang *et al.* [41] reviewed the recent developments in the design, methodology, and hardware platforms of environmental information systems for intelligent IoV. Liu *et al.* [42] provided a comprehensive overview of the introduction, architecture, challenges, application scenarios and other aspects of VEC. Patrikar *et al.* [43]

studied various methods and designs of anomaly detection in intelligent video surveillance. Besides, there are few studies on anomaly detection using edge computing, which needs attention.

#### 6) OPEN-SOURCE TOOLS AND SIMULATION PLATFORMS

Edge computing is a new paradigm that migrates networking, computing, and storage capabilities from remote clouds to the user side. In the context of IoT and 5G, open-source tools can generally reduce the data processing and transmission overhead and improve the efficiency and effectiveness of mobile data analysis. Liu *et al.* [44] classified the design requirements and innovations of open-source projects for edge computing systems as well as compared their goals, architectures, features, and limitations. In addition, energy efficiency improvement mechanisms and technological innovations for edge computing are investigated. On the other hand, the use of simulation platforms to implement for the study of systems is not only flexible but also economical, Van *et al.* [45] provided a comprehensive overview of edge computing simulation platforms and compared their characteristics.

#### 7) GENERAL OVERVIEW& OUTLOOK

Ahmed et al [46] detailed the classification in the field of MEC and the real-time application scenarios suitable for it, as well as identified the latest results and research challenges for successful MEC deployment.

In summary: Through a brief review of the latest survey, edge computing has gradually become a new direction of computing system, a new form of business in the information field and a new platform of industrial transformation by nearby providing key capabilities such as computing, network and intelligence. Based on the analysis of these survey articles, edge computing is in a high-speed development stage as a whole and is accelerating from concept popularization to pragmatic deployment, and it has attracted extensive attention by academia and industry.

### B. MOTIVATION OF THIS SURVEY

With the overview of existing edge computing-related surveys shown in Table 1, Table 2, Fig 1 and Fig 2, we have found that a survey of state-of-the-art on edge computing in terms of architectures and models, technology categories, research directions, and development paths and trends is still missing. To help better understand edge computing, we summarize multiple perspectives on architectures and models, technology categories, research directions, and development paths and trends. In particular, we attempt to include the latest high-quality research results that are not included in other existing survey articles, and we believe that this survey can provide new clues for further development of edge computing and provide researchers with a comprehensive, information rich and up-to-date view.

**TABLE 1. Taxonomy of existing edge computing-related surveys (Part 1).**

| Group | Category | Ref. | Year | Topic |
|---|---|---|---|---|
| Group-1: Edge Computing Essentials | Network Slicing | Rajkumar [1] | 2018 | Review on the interplay of 5G, edge/fog and cloud computing and aspects of network slicing |
| | | Shah[2] | 2021 | Researching MEC for 5G services and providing network slicing use cases |
| | Scalability | Pan[3] | 2018 | Focus on security and scalability for edge computing |
| | | Kumar[4] | 2021 | Discusses various approaches to improve the scalability and security of edge computing |
| | Security and Privacy | Alwarafy[5] | 2020 | Examining edge computing in IoT security and privacy issues |
| | Architecture | Abbas [6] | 2017 | Definition, advantages, architecture and application areas of MEC |
| | | Hamm[7] | 2019 | Edge computing architecture |
| Group-2: Resource Scheduling | Resource Management | Luong[8] | 2017 | Various incentives for resource management in cloud networks and shared resources in edge computing using pricing strategies |
| | | Hong[9] | 2018 | Discussion on future research directions and challenges for resource management in edge computing |
| | | Ghobaei [10] | 2020 | A classical classification of fog environmental resource management |
| | Computation Offloading | Mach[11] | 2017 | Focus on user-oriented computation offloading use cases in MEC |
| | | Wang[12] | 2018 | Research on key issues and methods related to edge-cloud offloading |
| | | Dziyaud din[13] | 2019 | Computation offloading techniques and content caching and delivery methods |
| | | Wang[14] | 2020 | The classification of task offloading in edge-cloud environments |
| | | Algarni[15] | 2021 | Study of different offload models for edge computing |
| | | Islam[16] | 2021 | The introduction of in-depth analysis and classification of task offloading in MEC |
| | Resource Provisioning | Duc[17] | 2019 | Investigation on the problem of reliable resource provisioning in federated edge-cloud environments |
| | | Spinelli [18] | 2021 | Exploration on the availability of resources for MEC and how it will enable the industry to grow vertically |
| Group-3: Decision-Making Techniques | Machine Learning and Deep Learning | Wen[19] | 2019 | Introduction to the emerging field of RRM of importance awareness |
| | | Mursheed[20] | 2019 | ML systems deployed at the edge of networks |
| | | Shakarami[21] | 2020 | A ML perspective based offloading method for mobile edge computing |
| | | Shuja[22] | 2021 | Study the ML techniques for caching within edge networks |
| | | Voghoei[23] | 2018 | Mapping DL to the edge computing paradigm |
| | | Wang[24] | 2020 | Providing the scenarios and basic enabling techniques for DL and MEC applicability |
| | | Sun[25] | 2021 | An overview of DDL challenges and solutions in edge computing |
| | Auction Method | Qiu[26] | 2021 | Auction and mechanism design in edge computing |
| | Game Theory | Moura[27] | 2018 | The Game theory applied to wireless resources on MEC services |

**TABLE 2. Taxonomy of existing edge computing-related surveys (Part 2).**

| Group | Category | Ref. | Year | Topic |
|---|---|---|---|---|
| Group-4: New Comm. Networking | 5G and Beyond | Taleb[28] | 2017 | MEC is a key emerging technology for 5G systems |
| | | Pham[29] | 2020 | An overview of the latest research on the integration of MEC with new technologies such as 5G |
| | | Al-Ansi[30] | 2021 | A survey and overview of intelligent edge computing technologies for 6G |
| | Blockchain | Yang[31] | 2019 | Integration of blockchain and edge computing system survey and overview |
| | | Nguyen [32] | 2020 | Recent research advances in the integration of blockchain with 5G networks and beyond |
| | | Queiroz [33] | 2020 | Outlines blockchain-based edge computing solutions for vehicles |
| | | Gadekallu[34] | 2021 | The introduction of the latest developments in BEoT technology |
| Group-5: Edge Computing Applications | Smart City | Khan[35] | 2019 | Application of smart city edge computing |
| | Smart Grid | Boccadoro[36] | 2018 | Application analysis of smart grid based on edge computing |
| | Industrial Internet | Li[37] | 2019 | Key technologies for MEC in industrial internet |
| | COVID-19 | Sufian[38] | 2020 | Edge computing research technology mitigates COVID-19 pandemic |
| | | Bianco[39] | 2021 | Using RFID systems for future pandemic outbreaks |
| | UAV | Zhou[40] | 2020 | Mobile edge computing in UAVs |
| | LOV | Zhang[41] | 2019 | Recent developments in environmental information systems of intelligent vehicles |
| | | Liu[42] | 2020 | A comprehensive overview of the latest VEC research |
| | Video Surveillance | Patrikar [43] | 2021 | Edge computing for anomaly detection system in video surveillance |
| Group-6: Platforms & Tools | Simulation Platforms & Open-Source Tools | Liu[44] | 2019 | An overview of existing edge computing systems and a comparison of open-source tools |
| | | Le[45] | 2021 | An overview of the e-commerce simulation platform is provided |
| Group-7: General Overview | Overview & Prospects | Ahmed[46] | 2016 | The introduction of the latest research results in the field of MEC |

### C. CONTRIBUTION OF OUR SURVEY

The survey introduced in this paper includes the following contributions: We briefly classify the existing edge computing surveys to emphasize the significance of the literature review shown in this survey, and then we make a comprehensive investigation of the latest architectures and models, key technologies, research directions and development paths of edge computing. Finally, several promising directions and problems to be solved in the future research are prospected.

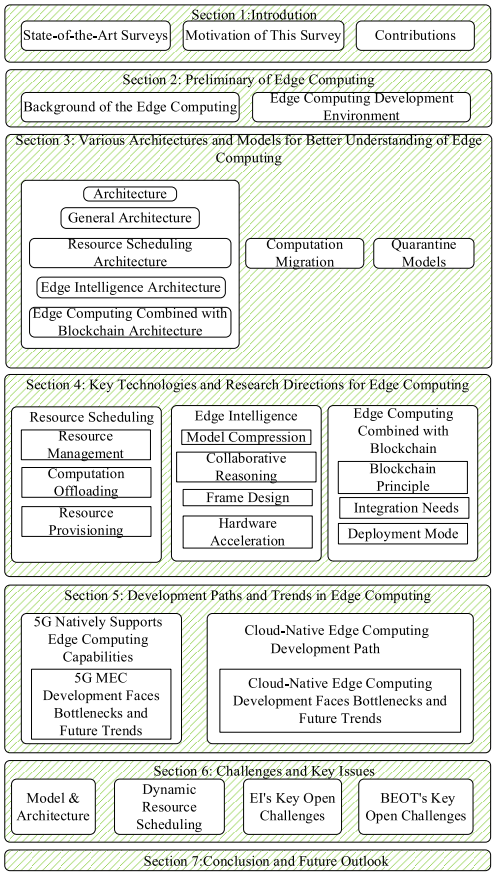


**FIGURE 3. The structure of this article.**

To help readers fully understand the structure of this survey, the structure of the study is shown in Fig 3 and organized as follows. Section 2 introduces the preliminaries of edge computing. Section 3 generalizes various architectures and models of edge computing. In Section 4, the latest technologies and research directions for improving the performance of edge computing are summarized. Section 5 describes the development paths and trends of edge computing. We look at the unresolved issues in Section 6. Finally, this article is summarized in Section 7.

## II. PRELIMINARIES OF EDGE COMPUTING

Edge computing is a new computing paradigm that provides applications with converged computing, storage and networking resources by being on the edge side of the network close to the source of things or data. In this section, we present the basic concepts, definitions and terminologies of edge computing appeared in this article. Due to the frequent use of acronyms in this paper, we will include an acronym table, i.e., Table 3, in this section.

### A. FOG COMPUTING AND EDGE COMPUTING

In the context of the Internet of Everything(IoE), data at the edge has seen explosive growth. To address the problems of computational load and bandwidth during data-oriented

**TABLE 3.** Acronym table.

| Acronym | Meaning | Acronym | Meaning |
|---|---|---|---|
| ADMM | Alternating Direction Method of Multipliers | GAN | Generative Adversarial Networks |
| AI | Artificial Intelligence | IoT | Internet of Thing |
| BS | Base Station | MD | Mobile Device |
| CAV | Connected and Autonomous Vehicle | MDC | Micro Data Center |
| CC | Cloud Computing | MDP | Markov Decision Process |
| CN | Core Network | MEC | Mobile Edge Computing |
| CNN | Convolution Neural Network | MIMO | Multiple Input Multiple Output |
| DNN | Data Network Name | NFV | Network Function Virtualization |
| DQN | Deep Q-network | QoE | Quality of Experience |
| DRL | Deep Reinforcement Learning | QoS | Quality of Service |
| D2D | Device to Device | SDN | Soft-defined Network |
| EC | Edge Computing | SCA | Successive Convex Approximation |
| EN | Edge Node | UAV | Unmanned Aerial Vehicle |
| ES | Edge Server | UE | User Equipment |
| FL | Federated Learning | VEC | Vehicle Edge Computing |
| FPGA | Field Programmable Gate Array | WAN | Wireless Access Network |

transmission, computation and storage, researchers have started to explore the addition of data processing at the edge close to the data producer, i.e., the uplink of the IoE service function. In 2012, Cisco proposed fog computing, defined as the migration of cloud computing centric tasks to network a highly virtualized computing platform executed by edge devices. The cloud computing architecture centralizes computing from the user side to the data center, keeping computing away from the data source, which can also bring problems such as computing latency, congestion, low reliability and security attacks. Fog computing(FC) is localized cloud computing, which is a complement to cloud computing. Cloud computing emphasizes more on the way of computing and fog computing emphasizes more on the location of computing. Provided that cloud computing is WAN computing, then fog computing is LAN computing.

Edge computing is not a new concept either. It first emerged in 2013, stemming from IBM and Nokia Siemens Networks then jointly launching a computing platform that could run applications inside wireless base stations to deliver services to mobile users. The European Telecommunications Standards Institute (ETSI) established the Mobile Edge Computing Specification Working Group in 2014 to officially announce a push for standardization of MEC. The basic idea is to migrate cloud computing platforms from inside the mobile core network to the edge of the mobile access network to achieve elastic utilization of computing and storage

**TABLE 4.** Acronym table.

| Features | CC | EC | FC |
|---|---|---|---|
| Delay | High | Low | Low |
| Calculating Resource Locations | Data Center | Edge Networks | Edge Networks |
| Communication Network | WAN | LAN, 4G/5G, etc. | LAN |
| Real-time | Low | High | High |
| Architecture | Centralized | Distributed | Distributed |
| Mobility | Low | High | High |
| Location Perception | Not supported | Supported | Supported |
| Security | Semi-secure connection | Safer | Safer |
| Types of services offered | Global-based services | Local information-based services | Local information-based services |
| Number of devices that can be served | Less | More | More |

resources. Since MEC is located inside the wireless access network and close to mobile users, it can achieve ultra-low latency and high bandwidth to improve service quality and user experience. With in-depth research, ETSI has further extended the definition of ''M'' in MEC to cover not only mobile access but also other non-3GPP access methods such as WI-FI access and fixed access, extending MEC from telecom cellular networks to other wireless access networks. ETSI redefined the ''M'' in MEC as ''Multi-Access,'' and the concept of ''Mobile Edge Computing'' was changed to ''Multi-Access Edge Computing.'' Edge computing and fog computing emerged to complement remote clouds to meet the service needs of a large number of geographically distributed IoT devices. The main difference between the two is that edge computing typically occurs directly on the device to which the sensor is attached or on a gateway device that is physically close to the sensor. Fog computing, on the other hand, shifts edge computing activities to processors connected to the LAN or to the LAN hardware itself, so they may be physically remote from sensors and actuators. Finally, Table 4 presents a comparison between their characteristics.

### *B. EDGE COMPUTING DEVELOPMENT ENVIRONMENT*

The statistical results of the literature ratio of the keywords ''edge computing'' and ''cloud computing'' through the search engine are shown in Fig 4. From the analysis of the development trend in recent years, cloud computing, as a mature and stable technical means, has been in a high-speed development stage since its emergence, but the proportion of literature tends to be flat and has basically reached saturation in recent years. In contrast, the proportion of edge computing is in progressive growth year by year since 2015, and the literature published in 2019 in one year accounts for more than half of the total number of the last seven years. It can be seen that edge computing is exactly the current focal technical issue.

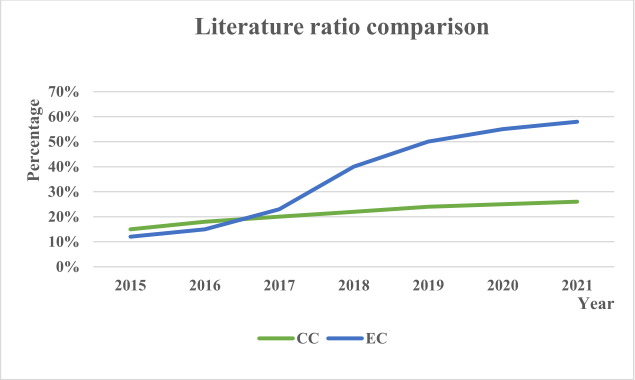


**FIGURE 4.** The comparison of the proportion of cloud computing and edge computing literature in recent years.

From 2015-2019, edge computing began to be vigorously promoted by the industry and entered a rapid growth phase, and 2019 was considered the first year of edge computing. With the rise of edge computing and the advent of the new digital era of IoE, the convergence and development of edge technology with cloud computing and other network technologies have become an important force in driving the implementation of edge computing technology, which have enabled the application of edge computing technology to steadily develop and enter industrial landing. Driven by the development of 5G, IoT and Industrial Internet, the global edge computing industry is booming. With the proliferation of applications and data volume, network bandwidth and computing throughput have become performance bottlenecks for computing. At the same time, the demand for real-time processing of massive amounts of ''small data'' generated by terminal equipment is growing at a high speed, driving edge computing to become an important computing platform for technology implementation in the data era, becoming a key support for agile connectivity, real-time services, and privacy protection in the digital transformation of the industry. It has become a key support for agile connectivity, real-time business and privacy protection in the digital transformation of the industry. According to CCID data, in 2020, the market size of edge computing reached 19.94 billion yuan, up 62.2% year-on-year; in 2021, the market size of China's edge computing reached 32.53 billion yuan, up 63.1% year-on-year.

In the future, as edge computing gradually enters a period of robust development, the development paths of 5G MEC, cloud-native edge computing, and vertical industry edge computing will develop in competition. This will present two prominent development features: 1) the value of building applications with single technologies of edge is difficult to realize, and it is necessary to combine edge computing with other technologies such as cloud computing, 5G, AI and blockchain to play synergistic effects and form integrated solutions; 2) standardization work, such as unified service definition, resource packaging, and interface protocols, will be continuously improved to promote efficient cooperation between different participants and cross-vendor product interoperability, so that the technology will gradually develop in the direction of open integration.

## III. VARIOUS ARCHITECTURES AND MODELS FOR BETTER UNDERSTANDING OF EDGE COMPUTING

### A. ARCHITECTURES

The architecture of edge computing is receiving increasing attention. Although several surveys [6], [7] have investigated edge computing architecture models as well as references to other architectural models, the current surveys have all focused on designing the architecture of edge computing platforms for specific computing scenarios. As edge computing is integrated into an increasing number of application domains, the appropriate model to describe the reference for this paradigm still seems to be undefined. In contrast, the architecture of edge computing in various scenarios is elaborated in our survey, which also summarizes the common architectures in edge computing. Whether it is a traditional computing scenario such as high-performance computing or an emerging computing scenario such as edge computing, the future architecture should be a model in which general-purpose processors and heterogeneous computing hardware coexist. Heterogeneous hardware sacrifices some of the general-purpose computing power and uses dedicated acceleration units to reduce the execution time of one or more types of loads and significantly improve the performance to power ratio. However, edge computing platforms are usually designed for a specific class of computing scenarios and handle a fixed type of load, so there is a lot of cutting-edge work to design edge computing platform architectures for specific computing scenarios.

#### 1) GENERAL ARCHITECTURE

This section describes the current general architecture of edge computing as shown in Fig 5. We outline the components of the architecture and introduce a three-tier heterogeneous edge computing network, in which the first layer is the thing layer, the second layer is the edge layer, and the third layer is the cloud layer, the components of which are described in detail below.

##### *a: THING LAYER*

Also known as user layer, it consists of various terminal equipment, such as IOVs, augmented reality devices, surveillance cameras, smart health sensors, etc.

##### *b: EDGE LAYER*

The edge layer is the middle part of this three-layer architecture, located at the edge of the network, and consists of a large number of edge nodes. Therefore, its hardware components usually include routers, gateways, switches, access points, base stations, specific edge servers, etc. From the aspect of software composition, the following functions are mainly realized: 1) routing subsystem can realize data forwarding

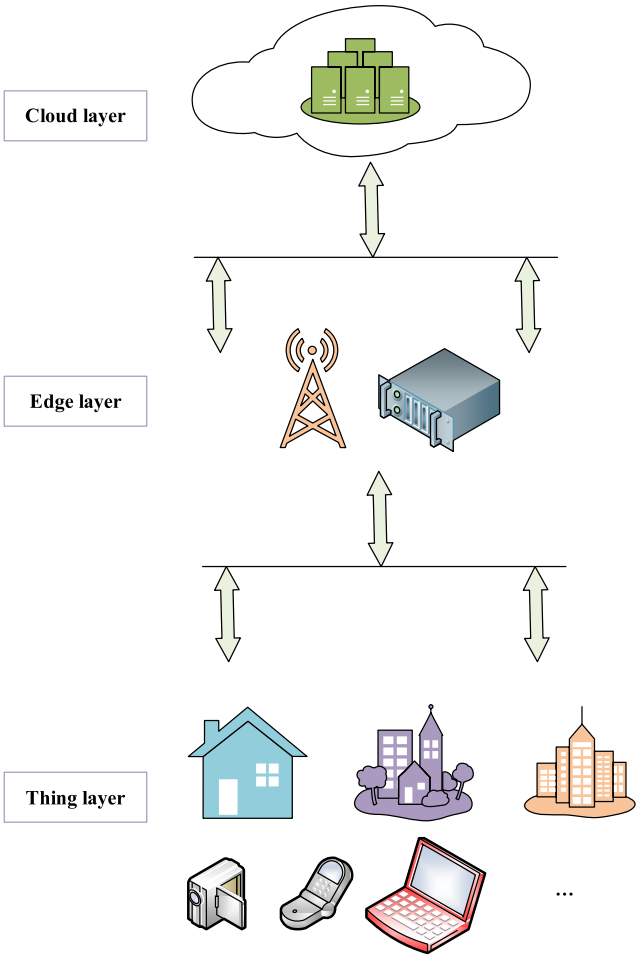


**FIGURE 5.** Edge computing architecture.

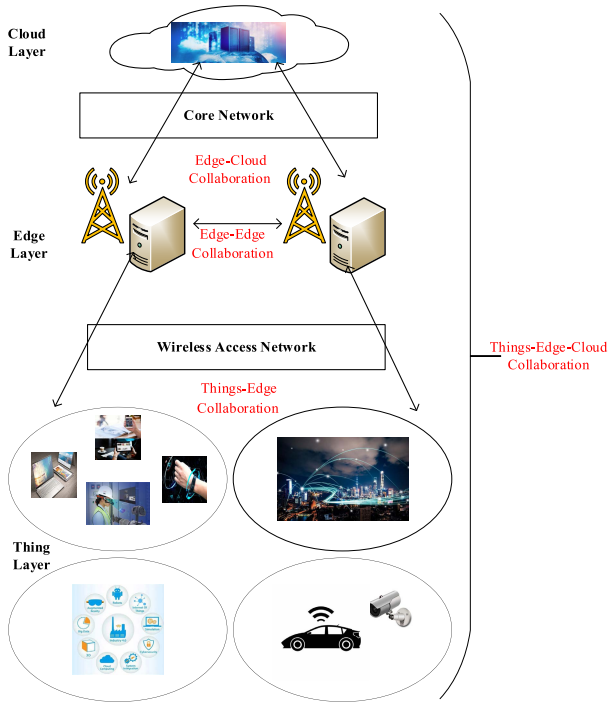


**FIGURE 6.** Resource scheduling architecture in edge computing.

and network connection, and support network virtualization; 2) capability open subsystem provides API interface and platform middleware to realize network capability invocation; 3) platform management subsystem mainly realizes control, infrastructure planning and scheduling, billing information statistical report, and other functions.

### *c: CLOUD LAYER*

The cloud layer has powerful data processing and storage capabilities. The current development trend of cloud layer is based on the core of cloud computing technology and the ability of edge computing, forming a comprehensive elastic cloud platform of computing, network, storage, security and other capabilities at the edge. The central cloud can form an E2E technology architecture with IoT endpoints for thing-edge-cloud collaboration, which has a placement of network forwarding, storage, computing, and intelligent data analysis in the edge, and a support of network-wide scheduling, arithmetic distributions and other cloud services.

The three-tier heterogeneous edge computing network architecture is accepted by many works of [47]–[49], etc. Ren *et al.* [47] then proposed a three-tier architecture and mechanism for edge computing to intelligently distribute computationally intensive tasks in AR applications to edge servers and cloud servers. In addition, typical image retrieval techniques are discussed, enabling further reduction in latency and energy consumption for mobile AR devices. Simulation results show that their schemes improve in performance metrics compared to existing schemes. To fill this gap in the edge computing paradigm for industrial applications, Willner *et al.* [48] introduced the RAMEC in the manufacturing domain. In addition, 210 views of this paradigm in the domain were identified, which provide the basis for future related research, standardization, and development activities. Rahimi *et al.* [49] proposed a hybrid design architecture for edge computing considering the needs of ultra-low latency applications and standardized deployments. The architecture makes use of state-of-the-art technologies enabling key features such as scalability, reliability, and ultra-low latency support. It is evaluated with agent-based simulations and the results show that it can achieve low latency response to high-capacity requirements.

### 2) RESOURCE SCHEDULING ARCHITECTURE

This section describes the edge computing architecture for resource scheduling. Based on the three-tier architecture, we propose different collaboration methods for resource scheduling of edge computing, as shown in Fig 6.

Things-Edge Collaboration: This collaborative processing of data generated by end devices can be processed locally or transmitted to an edge server, the choice being based on QoE and QoS requirements. For instance, Wang *et al.* [50]

optimized user rewards under network latency conditions based on a DEBO algorithm that handles task allocation involving many coexisting users in a dynamic and uncertain environment to meet the stringent user requirements for latency. Ale *et al.* [51] implemented computation offloading for multiple IoT devices and multiple ESs in a dynamic environment by developing D3PG based on the DDPG algorithm to describe the problem as a Markovian decision process with a constrained mixed action space. This increases the number of processing tasks and minimizes energy cost and service latency. Simulation results show that the D3PG algorithm outperforms existing schemes. Liang *et al.* [52] considered the sequential offloading of multiple users to one ES and theoretically proved the feasibility of the scheme. The comprehensive simulation results show that the sequence-optimized TDMA scheme has better throughput performance than the NOMA scheme.

Edge-Cloud Collaboration: Assuming that most of the computational tasks are executed in the cloud center, this will increase the load on the core network and the latency of task offloading, which will not meet the scenarios with high real-time requirements and QoE for users. Sinking computational tasks to ESs through edge-cloud collaboration will be a good approach. Su *et al.* [53] used two-tier Stackelberg game theory to MEC's application in the IoT market and proved its equilibrium existence. Then the game problem and negative utility among IoT MDs were solved using IPOA algorithm and ISPA algorithm. Experimental results show that these algorithms outperform traditional task offloading schemes. Li *et al.* [54] proposed a computational offload-based collaborative mechanism for migrating computational tasks to the edge and cloud, which handles the different computing tasks of the terminal by establishing a collaborative computation offloading model between cloud servers and edge servers. Experimental results show that the method provides significant improvements in reducing computational task execution time, improving server resource utilization, and reducing energy consumption of terminals compared to conventional optimization algorithms. In order to migrate tasks in dynamic environments from IoT devices to portable edge cloud servers, Qu *et al.* [55] proposed the DMRO algorithm, which can obtain the optimal offloading policy for complex tasks. Simulation results show that the algorithm has a significant improvement in offloading efficiency compared to the traditional DRL algorithm.

Things-Edge-Cloud Collaboration: While edge-cloud collaboration for IoT can solve most of the problems, with the increase of computationally intensive task programs, powerful cloud computing centers are needed to provide complementary computing resources. Therefore, a better realization of resource scheduling can be achieved through the things-edge-cloud collaboration approach. Ke *et al.* [56] considered a more general MEC scenario where UEs are distributed over a large area and multiple APs collaborate with each other to provide larger coverage. Tuli *et al.* [57] proposed a scheduling method called MCDS for efficient scheduling of workflow applications in mobile edge cloud computing systems. Comprehensive experiments show that the method promotes system improvements in terms of energy consumption, response time, and cost.

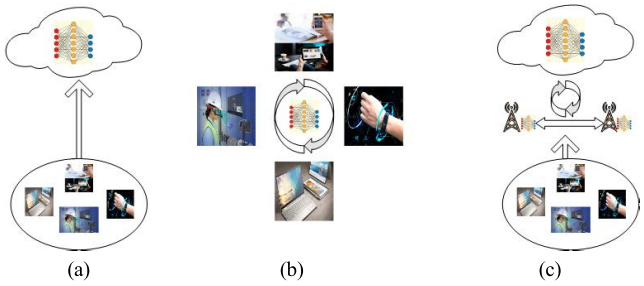


**FIGURE 7.** Architecture modes of distributed training. (a) Centralized; (b) Decentralized; (c) Hybrid.

Edge-Edge Collaboration: Thing-edge collaboration or thing-edge-cloud collaboration usually involves an edge-edge collaboration manner. He *et al.* [58] considered vehicles as edge computing nodes and proposed an online task offloading and resource allocation strategy based on the computing power of mobile vehicles that can handle the offloading task in vehicle-to-vehicle mode in a timely and efficient manner. To reduce the delay and allocation problems, the OPFTO offloading system and the improved HGSA algorithm are designed, respectively. Vehicle simulation experiments show that the strategy has the advantages of low latency and high accuracy. Based on the MEC architecture, Miao *et al.* [59] added an intelligent computation offloading strategy. To further reduce the total task latency, a predictive offloading strategy for computational tasks modelled on LSTM algorithm is also proposed. This strategy implements task scheduling between MD, ES and cloud and collaboration between ES.

### 3) EDGE INTELLIGENCE ARCHITECTURE

With the breakthrough of DL, Artificial Intelligence (AI) applications and services have boomed in recent years. Billions of mobile and IoT devices are connected to the Internet, generating trillions of bytes of data at the edge of the network. Driven by this trend, there is an urgent need to push the frontier of AI to the edge of the network to fully release the potential of edge big data. To meet this demand, edge computing, as a new model that pushes computing tasks and services from the network core to the network edge, has been widely recognized as a promising solution. The resulting interdisciplinary EI is starting to receive a lot of attention. In this section, we will focus on its architecture, which can be divided into three models: centralized, distributed, and hybrid, as shown in Fig 7.

#### *a: CENTRALIZED*

Fig. 7(a) depicts centralized training, where the model is trained in a cloud data center. The data used for training is generated and collected from distributed terminal equipment.

Once the data arrives, the cloud data center uses this data to perform the training. There are three inference models for training models in the cloud: 1) cloud–edge co-training and inference, which means that data is partially loaded into the cloud; 2) all in-edge, which means that model inference is per-formed within the edge of the network, which can be done by loading all or some of the data to the edge nodes; and 3) all on-device, which means that no data is loaded on the device.

#### *b: DISTRIBUTED*

In the distributed mode, as shown in (b) of Fig 7, each computing node trains its own model locally with local data, which keeps private information locally. In order to obtain the global model by sharing local training improvements, the nodes in the network will communicate with each other to exchange local model updates. In this mode, the global model can be trained without the intervention of the cloud data center to train and reason about the model in an edge-like manner.

#### *c: HYBRID*

Hybrid mode is a combination of centralized and distributed mode. As shown in Fig 7 (c), as the center of the architecture, the edge servers can train models by decentralizing updates to each other or by centralizing training through the cloud data center.

EI is mainly concerned with model compression and collaborative reasoning. With the popularity of cell phones and other terminal equipment and the increase in computing power, as well as advances in AI, many smart applications have been developed to enrich people's lives. Many smart applications rely on DL models, such as CNNs. To improve the performance of these methods, the trend is to use increasingly deeper architectures and more parameters, which leads to higher computational costs. Considering the limited computing resources and energy consumption at the edge, how to efficiently deploy DL models at the edge is a very interesting problem to investigate. The current EI inference models include as shown in Fig 8: 1) edge-based model, where the model inference is performed on the edge server and the predictions are returned to the device; 2) device-based model, where the mobile device acquires the model from the edge server and performs the model inference locally; 3) edge-based device model, where the device executes the model to the specified layer and then sends the intermediate data to the edge server, the edge server executes the remaining layers and sends the predictions to the device; 4) edge-cloud mode, where the device is mainly responsible for input data acquisition and the model is executed on the edge and cloud.

### 4) EDGE COMPUTING COMBINED WITH BLOCKCHAIN ARCHITECTURE

Blockchain and edge computing are both based on having the same distributed mechanisms in computing, data storage and networking, as well as their different and complementary focus, predestining them to be combined. On the

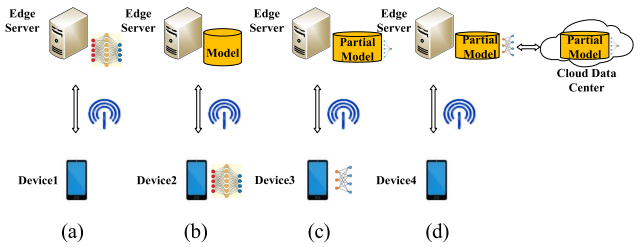


**FIGURE 8.** EI inference models. (a) Edge-based mode; (b) Device-based mode; (c) Edge-device mode; (d) Edge-cloud mode.

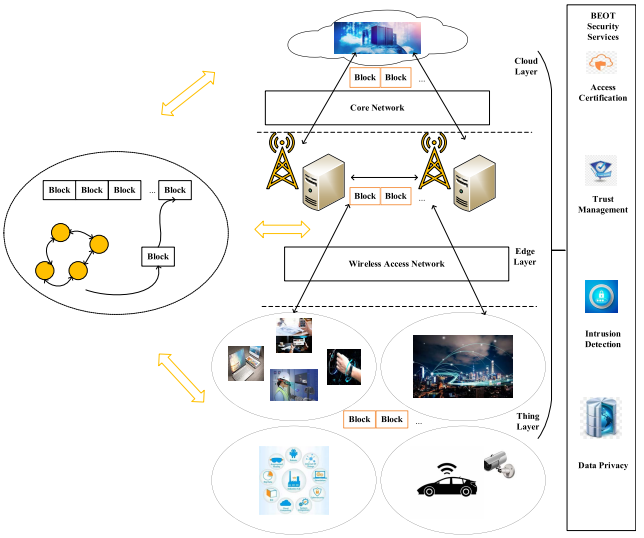


**FIGURE 9.** Blockchain and edge of things architecture.

one hand, the integration of blockchain into edge computing enhances security, privacy and automated resource usage. Using blockchain technology, it is possible to build distributed control on dozens of edge nodes and also to protect the accuracy, consistency and validity of data and rules in a transparent way. Thus, it is an effective solution for large numbers of heterogeneous users moving at or between physical edges. On the other hand, the integration of edge computing into blockchain brings a powerful distributed network and abundant compute and storage resources at the edge of the network. Using edge computing, computation can be moved from end devices to edge servers, enabling end users with limited resources to participate in the blockchain.

We propose a novel blockchain and edge of things (BEoT) architecture that is implemented by using blockchain in resource scheduling, as shown in Fig 9. The architecture consists of four main entities: terminal equipment, edge computing, blockchain, and cloud center.

## *B. COMPUTATION MIGRATION*

Edge computing organically integrates computing, storage and other resources on the edge of the network to build a unified user service platform that responds to and effectively handles task requests from network edge nodes in a timely manner according to the proximity service principle. Due to the limited capacity, resources, bandwidth, and energy of

edge nodes, computational migration becomes exceptionally important. In mobile edge computing, local edge servers can host cloud-based services, which reduces network overhead and latency, but service migration is required as users migrate to new locations. Optimizing migration decisions is challenging due to the uncertainty in dynamic cloud environments. The rise of computational migration techniques has introduced new approaches to address the resource constraints of mobile terminals. Chen *et al.* [60] proposed an ECC network architecture and a dynamic service migration mechanism to achieve low latency and user behavior prediction. The results show that the architecture and mechanism can better guide service migration in edge computing environments. Chang *et al.* [61] proposed a REM scheme for optimal process migration decisions when mobile sensors migrate tasks to multiple heterogeneous FEC resources and developed the EPIoT host framework. Experimental results show that the scheme and framework can improve the performance of heterogeneous process migration in FEC environments. Yousafzai *et al.* [62] proposed a lightweight PMCO framework for MEC with process migration support, which enables seamless migration of native applications and resource-intensive IoT application processing. The results show that the framework can significantly improve time efficiency and energy efficiency. Ngo *et al.* [63] proposed a MEC architecture and a coordinated migration switching mechanism to address the joint problem of container migration and base station switching as well as to reduce E2E latency. The results of this mechanism are shown to help improve the user experience. Wang *et al.* [64] proposed a learning-driven method named DRACM that can make effective online migration decisions in highly dynamic environments and user mobility. The approach provides improvements in terms of scalability and latency. Numerous experimental results show that the authors' approach outperforms traditional algorithms. Liang *et al.* [65] proposed an optimal migration/switching strategy between BSs that used relaxation and rounding efficient solution methods to solve complex combinatorial problems. In addition, for over-loaded BSs, the load can be migrated to a nearby idle BS. From the simulation results, this migration strategy can reduce the migration cost and improve the offloading efficiency. Xu *et al.* [66] developed a PDMA approach to address the problems of latency and mobility management in MEC environments. The results derived on the iFogSim platform show that the method improves user experience, improves performance by 8% - 20% and reduces migration costs by more than 75% during urban peak traffic hours compared to the baseline scheme. Goudarzi *et al.* [67] proposed a new weighted cost model and a distributed migration management technique to minimize operational costs and migration costs. The results show that their technique is able to improve in terms of deployment time, migration cost, total number of migrations, and total number of disrupted tasks. To address the problems of large-scale scenarios in edge computing that bring complexity in network topology and rapid increase in migration request, He *et al.* [68] proposed iterative MIS-based algorithms, which can effectively schedule multiple dynamic container migrations in this environment. Experiments show that it improves a lot in terms of processing time and migration performance compared to cloud migration planning algorithms.

### C. QUARANTINE MODEL

In cloud computing scenarios, a crash of one application can bring instability of the entire system with serious consequences. And in edge computing, the situation becomes even more complex. For example, in autonomous driving, it is necessary to support in vehicle entertainment while also satisfying user driving requirements. At this point, if the two tasks interfere with each other, the safety of autonomous driving will be seriously affected. Compared with the cloud computing scenario where resource quarantine is guaranteed by using VM and Docker technologies, edge computing can learn from its experience and study quarantine techniques suitable for edge computing scenarios.

Quarantine techniques are research tool to support the robust development of edge computing, through which the edge devices gain reliability of service and quality of service. Quarantine techniques need to consider 2 aspects: 1) quarantine of computational resources, i.e., applications should not interfere with each other; 2) quarantine of data, i.e., different applications should have different access rights. Ha *et al.* [69] proposed a VM switching technique that enables VMs migration for computational tasks, supports rapid placement of services, and ensures encapsulation of VMs in applications with high security and manageability requirements. In addition, this dynamic migration feature and quarantine techniques allow optimization of the edge side and increases the availability of the edge computing system. Mahadevappa *et al.* [70] proposed a new concept of data quarantine model that guarantees the integrity of data in edge computing. Specifically, the model isolates the identified data for a predefined period of time and does not cause data from adjacent edge nodes.

In summary: The design of computing system architectures for edge computing is still an emerging field and still has many challenges that need to be addressed.

## IV. KEY TECHNOLOGIES AND RESEARCH DIRECTIONS FOR EDGE COMPUTING

Currently, academic research on edge computing focuses on advanced scheduling strategies and technologies such as resource scheduling, EI, and edge computing combined with blockchain. From the analysis of these examples, whose features are summarized in Table 5, Table 6, Table 7, and Table 8.

### A. RESOURCE SCHEDULING

#### 1) RESOURCE MANAGEMENT

The integrated management of wireless and computational resources is an important part of the MEC system design. For different MEC system setups, we need to

**TABLE 5. Edge computing focuses on the comparison of the latest resource management articles.**

| Ref. | Research Content | Use of technology or algorithms | Performance Indicators |
|---|---|---|---|
| Dlamini[71] | Adaptive resource management for virtualization in EC | Propose an automatic resource control algorithm for energy-aware servers | Energy consumption and cost |
| Wan[72] | Path planning and resource management of UAV base stations based on MEC | Online edge processing scheduling algorithm based on Lyapunov optimization | Delay and energy consumption |
| Jošilo [73] | Combining wireless and EC resource management with dynamic network slicing options | Improving algorithm based on game theory | Delay |
| Chen[74] | Cooling-aware resource allocation and load management for MEC | Resource allocation and load management algorithm for cold-aware WPT-MEC systems | Energy consumption |
| Zeng[75] | Energy-efficient resource management of federated edge learning with heterogeneous computing | FL | Delay and energy consumption |
| Yu[76] | Ultra-dense EC | DRL and FL | Latency and resource utilization |
| Moro[77] | Joint management of compute and radio resources in MEC | Convex optimization and game theory | Service utility |
| Zaw[78] | Energy-aware resource management for federated learning in multi-access MEC | FL | Delay and energy consumption |

**TABLE 6. Edge computing focuses on the comparison of the latest computation offloading articles.**

| Ref. | Research Content | Use of technology or algorithms | Performance Indicators |
|---|---|---|---|
| Yan[79] | Optimal task offloading and resource allocation for MEC with inter-user tasks | Bi-section search method and gibbs sampling algorithm | Delay and energy consumption |
| Liu[80] | Dynamic task offloading and resource allocation for MEC | Lyapunov optimization and matching theory | Delay |
| Sun[81] | Adaptive learning task offloading for VEC | ALTO algorithm | Delay |
| Du[82] | Business-enhancing task offloading and resource allocation in multi-server MEC | Online OJTORA algorithm based on Lyapunov optimization | Delay and energy consumption |
| Wang[83] | Optimal energy allocation and task offloading strategies for wireless MEC | Convex optimization | Energy consumption |
| Batewela[84] | Risk-sensitive task seizure and offloading for VEC | Distributed no-regret learning algorithm | Delay |
| Lucic[85] | A latency-aware task offloading in MEC network for distributed elevated LiDAR | Constructing a mixed-integer programming problem | Delay |
| Tang[86] | DRL for task offloading in MEC | DRL | Delay and task drop rate |
| Huang[87] | VEC network for computational task offloading and resource allocation | JORA-MADDPG algorithm | Cost, delay and energy consumption |
| Sun [88] | A game-theoretic approach to resource allocation and task offloading in VEC | Game theory | Performance and efficiency |
| Li [89] | A Contract-Stackelberg approach to resource allocation and task offloading in VEC | Stackelberg game | Profit |

address different integrated resource management problems. Dlamini *et al.* [71] considered a computation and communication energy model and proposed an online server management algorithm called ARCES that minimizes the total energy consumption and cost. Numerical results show that ARCES saves an average of 69% energy relative to the case without energy management. Wan *et al.* [72] proposed a three-tier online data processing network based on the MEC technique, and developed a network scheduling algorithm and an online path planning algorithm based on Lyapunov optimization. Simulation tests show that the system improve data latency, power consumption, and service coverage. Jošilo *et al.* [73] studied a network slicing-based edge computing system that solves the JSS-ERM problem with an approximation algorithm with a bounded approximation ratio, which reduces the computational complexity. Test results show that the system is able to achieve an increase in performance compared to no-slicing and equal-slicing. Chen *et al.* [74] proposed a cooling-aware WPT-MEC system that minimizes the total energy consumption without reducing the latency requirements of the AP by the alternating optimization technique and Lagrange duality method to co-design its resource location and load management. Extensive numerical experiments show that the scheme can save nearly double the energy consumption compared to the baseline scheme. Zeng *et al.* [75] designed a framework named $C^2RM$ and further designed a device scheduling scheme and a greedy spectrum sharing scheme based on it, which improves performance and energy efficiency for joint management of computational communica-

**TABLE 7. Edge computing focuses on the comparison of the latest resource provisioning articles.**

| Ref. | Research Content | Use of technology or algorithms | Performance Indicators |
|---|---|---|---|
| Abouaomar[90] | EC resource issuance for latency-sensitive applications | Lyapunov optimization | Delay and energy consumption |
| Tilahun [91] | Resource Scheduling for MEC Based on RL | MADDPG algorithm | Energy consumption |
| Ascigil[92] | Resource allocation for the function-as-a-service edge- cloud | A similar approach for FaaS edge-clouds | Delay |
| Nasimi[93] | Auxiliary congestion control mechanism at the edge of 5G networks for SDN | Congestion control algorithms | Delay |
| Gao [94] | Distributed virtual network function arrangement method in satellite edge and cloud computing | D-VNFP algorithm | Delay |
| Ly[95] | Data compression and task allocation for MEC | Convex optimization | Delay and energy consumption |
| Yan[96] | A game theory-based algorithm for joint task offloading and resource allocation for MEC | Game theory | Delay and energy consumption |
| Zhong[97] | Parallel optimal task allocation mechanism for large-scale MEC | ADMM approach | Delay and energy consumption |
| Zhang[98] | Privacy-aware task assignment in social sensing-based EC | Game theory | Delay and payoff |

tion resources. In addition, the authors conducted experiments using real datasets to verify that the framework can improve the energy efficiency of FEEL systems. Yu *et al.* [76] proposed a framework called I-UDEC and a method called 2TS-DRL for jointly optimizing the computation offloading, allocation location, and cache placement problems of resources in UDEC networks to minimize the total offloading delay and resource utilization. Simulation results validate that the method can reduce the task execution time up to 31.87% under this framework. Moro *et al.* [77] designed an allocation mechanism using convex programming that solves the management problem of computing resources and radio resources in a MEC system, guaranteeing efficiency and fairness for the system. The simulation results confirm the theoretical nature of the market model. Zaw *et al.* [78] proposed a MEC-enabled FL model and an energy-aware resource management algorithm that addresses the balance between performance and device energy consumption, as well as reducing global polling and time consumption. Simulation

**TABLE 8. Edge computing focuses on the comparison of the latest model compression articles.**

| Ref. | Research Content | Challenges | Impact | Technology |
|---|---|---|---|---|
| Gamanayake[99] | An effective pruning method for edge vision filters | CNN applications in the field of computer vision | Quantitative irregularities affect neural computing hardware architectures | Cluster pruning |
| Qian [100] | Optimize task offloading and resource allocation in MEC | Use mixed integer offload policies to aid in resource allocation for task offloading | High calculation volume | IBNB |
| Libri[101] | Real-time malware detection for data centers with EI | "Big Data" stream support | Anomaly detection and security analysis | New lightweight and scalable approach to pAElla |
| Huang[102] | Resource-constrained training of EI | Neural network training on edge terminals | Compact model design | RCT |
| Subedi[103] | Parallel DL model execution and dynamic model placement on edge devices | Performance of edge devices performing multiple DL tasks | DL Inference Service for Edge Deployment | Parallel model execution and dynamic model placement |
| Liu[104] | Optimized gradient quantization and bandwidth allocation for minimizing training time | Training ML models using FEEL is time consuming | The cost is to increase the communication rounds | SCA |
| Chakraborty[105] | Hybrid network architecture for extreme quantized neural networks | Finding efficient implementations of neural networks in computing and storage | Causes significant performance degradation | PCA-based one-shot method |
| Gorsline[106] | Adversarial robustness of quantitative neural networks | Reducing the size of neural network models | The effect of quantization on the robustness of neural networks against | Quantization techniques |
| Liu [107] | On the initialization method of neural network weights with asymmetric activation function | Expand the choice of activation function and speed up the convergence of the network | Gradient vanishing problem | GLIT |
| Chen [108] | Quantitative acceleration of training with EI | Development and implementation of edge-oriented AI algorithms (e.g., accelerator) | Limiting the effectiveness of DNN algorithms | VecQ quantization method and T-DLA design |
| Chen [109] | Compression updates DNN models on edge devices | Updating DNN models on edge devices | Limited by bandwidth and equipment | Matrix factorization |
| Andreev [110] | Quantification of GANs | Deployment on edge devices gets complicated | Complicates the reasoning of the hardware | Quantization techniques |

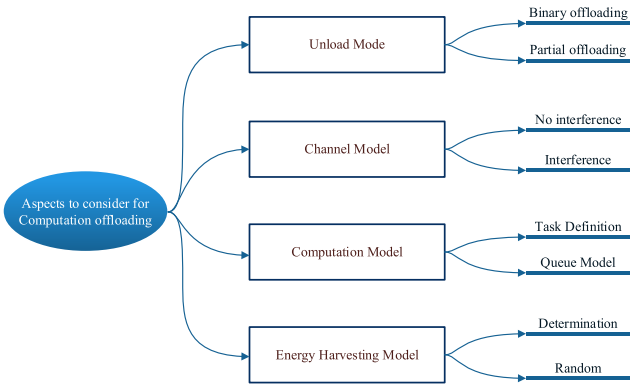


**FIGURE 10.** **Aspects to consider for offloading models for edge computing.**

results show that the model has a lower total time consumption compared to the traditional FL method.

### 2) COMPUTATION OFFLOADING

To cope with the problems of insufficient processing power and limited resources of end devices, the industry has introduced the concept of computation offloading in MEC. Edge computing offloading, that is, the unified equipment (UE) offloads the computing task to the MEC network, which mainly solves the shortcomings of the equipment in resource storage, computing performance and energy efficiency.

#### *a: UNLOAD MODE*

Binary offloading has only a difference between 0 and 1, which means that the offloaded tasks are packed and cannot be split. In contrast, partial offloads allow to partition the task and execute a task by first dividing it into different components and then making the corresponding offload decisions for those components. In practical problems, binary offloading is more common because we usually consider simple non-separable tasks, while partial offloading is applicable to complex tasks with multiple segments of tasks in parallel.

#### *b: CHANNEL MODEL*

For the interference-free channel model, either time division multiple access (TDMA) or orthogonal frequency division multiple access (OFDA) techniques can be adopted, and the transmission rate can be calculated as

$$r_k = B\log_2(1+\frac{p_k h_k^2}{N_0}) \tag{1}$$

This is the Shannon formula, where $B$ is the channel bandwidth (Hz); $N_0$ is the Gaussian noise power inside the channel; $p_k$ is the transmitted power of the end devices, and $h_k$ represents the channel gain.

For the channel model with interference, the code division multiple access technique can be used, while the calculation of the transmission rate is expressed as

$$rk = B\log_2(1+\frac{p_k h_k^2}{N_0+\sum\limits_{i\neq k} p_i h_i^2}) \tag{2}$$

where $\sum\limits_{i\neq k} p_i h_i^2$ is a reflection of the mutual interference between different devices. If both are considered, it is called a hybrid channel model.

#### *c: COMPUTATION MODEL*

Computational tasks have different definitions, the first one is $(\omega_k, s_k)$, where $\omega_k$ denotes the number of CPU cycles required to complete the computational task and $s_k$ denotes the size of the computational input data. The computational latency and energy consumption for local execution are defined as

$$t_k^L = \frac{\omega_k}{f_k}, \quad e_k^L = \rho\omega_k \tag{3}$$

where $t_k^L$ and $e_k^L$ represent the latency and energy consumption due to local computation, respectively. Then $f_k$ is the modeling of the computing power of the end devices and it is the number of CPU cycles per second executed by the device. The energy consumed in this process is proportional to the number of CPU cycles with a scale factor of $\rho$. In addition, the transmission of data also causes latency and energy consumption, denoted by$t_k^J$ and $e_k^J$, respectively.

$$t_k^J = \frac{s_k}{r_k}, \quad e_k^J = p_k \cdot t_k^J \tag{4}$$

where $r_k$ is the communication transmission rate focused on in Equation 1, and $p_k$ represents the data transmission power of the terminal device.

Another way of defining the computational task is $(s_k, \tau_k)$, where $\tau_k$ portrays the tolerance, i.e., the latency requirement. The computational latency and energy consumption of the local execution are as

$$t_k^L = \frac{\alpha \mathrm{s}_k}{f_k}, \quad e_k^L = \alpha \mathrm{s}_k \beta f_k^2 \tag{5}$$

In contrast to the previous one, instead of directly introducing the number of CPU cycles, using the proportionality between the number of CPU cycles and the data size $\mathrm{s}_k$, where the scale factor is $\alpha$. The energy consumed per CPU cycle is proportional to $f_k^2$, and the scale factor is $\beta$.

In addition, we can introduce the knowledge of queuing theory, using the queuing model, the queue in the terminal device modeled as $M/M/1$, the task arrival interval is taken as a negative exponential distribution, the service time is a negative exponential distribution, the average task generation rate is $\lambda$, the service rate is $u$, so the resulting delay calculation and energy consumption are as

$$t_k^L = \frac{\frac{1}{u_k}}{1-\frac{\lambda_k}{u_k}}, \quad e_k^L = \gamma\lambda_k\tau \mathrm{s}_k \tag{6}$$

where $\gamma$ is the energy consumed per bit, and $\lambda_k \tau$ is the total number of tasks arrived in each time period. In addition, the delay and the energy consumption in the transmission process are as

$$t_k^J = \frac{\lambda_k \tau s_k}{r_k}, \quad e_k^J = p_k t_k^J \tag{7}$$

Energy Harvesting Model: Energy supply equipment is very stable and the amount of energy that can be supplied per unit of time is known and predictable, which is the deterministic model. However, this is often unrealistic in real life. Therefore, most studies on energy harvesting models have focused on random models. In addition, some researchers have proposed bimodal models of energy states and environmental states. The energy $H(\mathrm{t})$ arriving in each period can be derived from the environmental state $e(t)$. $H(\mathrm{t})$ is modeled as a random variable given the state $e(t)$, obeying a conditional distribution, denoted as

$$P_H(H(\mathrm{t})|e(t)) \tag{8}$$

The relevant literatures on computation offloading are as follows. Yan *et al.* [79] investigated a mixed-integer optimization problem and a bi-section search method to minimize the energy consumption and task execution time of WDs under a task-dependent model. In addition, the Gibbs sampling algorithm was proposed to obtain the optimal unloading decision based on the one-climb strategy. Simulation results show that the method has higher performance than the benchmark algorithm. Liu *et al.* [80] studied a task offloading and resource allocation framework with URLLC and a user-server association method. In addition, Lyapunov is used to optimize the stochasticity of situations such as task arrival as well as to correlate UE with MEC servers on long time scales using matching theory. The simulation results show that the authors' proposed partial offloading scheme has more reliable task execution compared to the no-MEC-server and full offloading schemes. Sun *et al.* [81] designed an adaptive learning based ALTO algorithm to address task offloading in VEC systems. Simulation results show that this algorithm reduces the average latency by 30% compared to the traditional upper confidence bound algorithm. Du *et al.* [82] proposed an online algorithm based on Lyapunov optimization, OJTORA, to solve multi-server joint task offloading and resource allocation, and then to transform them into a deterministic optimization problem within each time slot. Experiments show that the authors' method outperforms the baseline method in terms of service capacity and service cost, but the algorithm does not investigate the dynamic allocation of bandwidth. Wang *et al.* [83] investigated a single-user wireless powered MEC system using convex optimization techniques to enable energy minimization. Next, heuristic algorithms were designed for WPT and task assignment on users. Compared the scheme of author with the benchmark scheme, it is shown that the energy consumption of the proposed scheme is much lower, and the performance of the proposed algorithm is close to the offline optimal solution. Bate-wela *et al.* [84] proposed a distributed no-regret learning algorithm based on a risk-sensitive task grasping and offloading scheme to solve the ultra-reliable low-latency communication problem in VEC networks. Simulation results show that compared to other baseline schemes, the authors' scheme has a reduction in latency. Lucic *et al.* [85] proposed ELiD as an alternative to local LIDAR sensors for AVs and achieved minimization of the average delay by constructing a mixed-integer programming problem. The results show that the scheme improves the utilization of the network and the robustness of the system. Tang *et al.* [86] proposed a DRL-based task offloading algorithm for the task offloading problem of an indivisible and delay-sensitive MEC system. Simulation results show that the authors' proposed algorithm has reduced task drop rate and average delay than other online offloading schemes. Huang *et al.* [87] proposed a joint offloading and resource allocation algorithm for JORA-MADDPG, which solves the problem of task type and vehicle speed constraints on task delay. Simulation results show that the algorithm has good results in terms of delay, energy consumption and efficiency. Sun *et al.* [88] proposed a method, GTRATOP, in order to optimize the VEC network, which solves the resource allocation and task offloading problem of this network. Simulation results show that the method has better performance and efficiency in the case of system reloading. Li *et al.* [89] proposed a multi-stage Stackelberg game under a contract-based incentive mechanism to deal with the idle resource problem of vehicles. Simulation results show the effectiveness of the scheme.

### 3) RESOURCE PROVISIONING

Workloads in MEC environments often experience frequent fluctuations in load uncertainty due to very frequent request events occurring on the thing side. These uncertainties can lead to resource provisioning issues. Abouaomar *et al.* [90] proposed a resource representation model and a Lyapunov optimization-based resource allocation scheme. The former is able to represent various resources of EDs and the latter minimizes the latency. Simulation results show that the proposed scheme by the author is superior to other benchmark schemes in terms of latency and energy consumption metrics. Tilahun *et al.* [91] proposed a JCCRA problem and a MADDPG algorithm, which solves the task to minimize the user's energy consumption while satisfying tight delay constraints. Simulation results show that the authors' proposed scheme significantly outperforms the heuristic baseline in terms of energy consumption. Ascigil *et al.* [92] proposed a heuristic algorithm in resource allocation and configuration considering edge computing for FaaSaaS. Simulation results show that the policy can achieve almost the same performance without coordination or communication overhead compared to a fully centralized policy. Nasimi *et al.* [93] proposed a congestion control mechanism that works within the MEC framework and an edge-assisted congestion control scheme. The former can effectively alleviate network congestion under different network conditions and QoS and

can selectively offload traffic based on real-time decisions. Simulation results validate the performance improvement of the proposed scheme. Gao *et al.* [94] established a hierarchical satellite edge and cloud computing framework and proposed an algorithm called D-VNFP. The results show that the algorithm has better performance in terms of satellite network bandwidth consumption and service E2E delay compared to algorithms Greedy and Viterbi. Ly *et al.* [95] envisioned a mobile edge offloading scenario and proposed a computation offloading framework and a method to jointly optimize task assignment decisions and data compression ratios. The authors transformed this approach into a convex optimization problem and achieved the goal of minimizing energy consumption and latency. Simulation results show that the scheme compares favorably with the benchmark scheme at low data rates. Yan *et al.* [96] proposed a game theory-based approach to reduce the energy consumption and delay of multi-user collaborative offloading decisions and resource allocation in MEC systems. Simulation results also show that this algorithm can help the performance of multi-user MEC systems. Zhong *et al.* [97] proposed a method called ADMM to express the joint optimization problem of delay and energy consumption metrics in MEC systems as a nonlinear 0-1 integer programming problem. A series of algorithms (yoke gradient, logarithmic smoothing, etc.) were used to solve this optimization problem. The simulation results show that the mechanism achieves the joint optimization purpose and effectively reduces the delay and energy consumption of this system. Zhang *et al.* [98] investigated a game theory-based G-PATA framework to solve the privacy-aware computational task allocation problem in SSEC systems. The results show that G-PATA reduces latency nearly to half over the baseline solution on applications with various privacy settings. It also improves the gain of the end devices by a factor of 0.15.

### B. EDGE INTELLIGENCE

With technologies such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Generative Adversarial Networks (GAN), and Deep Reinforcement Learning (DRL) in the field of AI, edge computing can flourish with richer data and application scenarios. As a result, the combination of edge computing and AI has given rise to a new research area called ''edge intelligence.'' EI leverages a wide range of edge resources to support AI applications without relying exclusively on the cloud. While edge AI or EI is a completely new term, practice in this direction began early, with Microsoft building an edge-based prototype to support mobile voice command recognition in 2009. However, despite the beginning of early exploration, EI still has no formal definition. Currently, most organizations and media refer to EI as ''a paradigm for running AI algorithms locally on end devices where the data (sensor data or signals) is created on the device.'' EI promises to make it possible to develop a variety of distributed, low-latency, and reliable intelligent services. As we learned from the introduction of the architecture of EI models in Chapter 3, the current main research directions of EI are divided into four areas, namely, model compression, collaborative reasoning, framework design, and hardware acceleration.

#### 1) MODEL COMPRESSION

The primary motivation for pushing learning to the edge is to allow rapid access to the vast amount of real-time data generated by edge devices for rapid AI model training and inference, thus giving the device human-like intelligence to respond to real-time events. Because many AI applications require high computational power, this greatly exceeds the capabilities of resource and energy-constrained edge devices. Therefore, enabling models to run on more lightweight edge computing devices generally allow for compression operations such as distillation, pruning, and quantization to reduce memory and computation. Gamanayake *et al.* [99] proposed a cluster pruning method that prunes the entire network by collaborating the underlying hardware structure parameters and a greedy algorithm to determine the optimal cluster size. This method has advantages in terms of latency and accuracy compared to traditional filter pruning algorithms. Qian *et al.* [100] proposed a pruning strategy, IBnB, which guarantees structurally near-optimal performance and reduces the complexity. Simulation results show that the method has good performance. Libri *et al.* [101] proposed a method called pAElla to malware detection in real time. The results show that in DCs / SCs environment, pAElla can cover a wider range of malware and improve the accuracy compared to the SoA method. Huang *et al.* [102] proposed a method called RCT for the edge devices capacity problem. On the one hand, this method preserves the quantization model by during the training process. On the other hand, the bit width per layer can be dynamically adjusted. It both reduces the memory requirement of parameters and saves energy. Experimental results show that the RCT method outperforms other methods (e.g., GEMM and QAT). Subedi *et al.* [103] investigated two techniques, parallel model execution and dynamic model placement, to determine the benefits and limitations of AI multi-tenant models such as image classification on edge devices. The results of simulations on Jetson TX2 show that this scheme improves DL inference throughput by a factor of 3.3 to 3.8. Liu *et al.* [104] proposed a training time model and an alternating optimization-based algorithm to solve the training time minimization problem in the quantized FEEL system. Experiments show that the optimization algorithm proposed by the authors can approach the optimal performance under different learning tasks and models. Chakraborty *et al.* [105] proposed a PCA-based one-shot method. This method is used to design hybrid compressed neural networks and identify the important layers of binary networks, offering the possibility of using energy-efficient hybrid networks in low-power edge devices. Simulation results show that the accuracy of the scheme is close to that of a full precision network. Gorsline *et al.* [106] explored geometric models of the robustness of quantized neural networks with different dimensions and different activation functions

against gradient attacks. Simulation results show that for simple gradient-based attacks, quantization can improve or reduce the robustness of the countermeasures depending on the attack strength. In addition, Liu *et al.* [107] proposed an improved method for initializing the weights of asymmetric activation function neural networks, which expands the selection range of activation functions and improves the performance of the network. Chen *et al.* [108] proposed a neural network model and a quantification method for VecQ. The model is able to reduce a large amount of memory requirements as well as the method achieves state-of-the-art accuracy at the same compression ratio. In addition, the authors investigated an accelerator design named T-DLA for DNNs. Chen *et al.* [109] proposed a DNN layer-based reparameterization method to update compression model. Simulation results show the superiority of the authors' method over existing update compression techniques in terms of update size and on inference accuracy. Andreev *et al.* [110] conducted an experimental study of post-training quantization and quantization-aware training techniques for three different GAN structures and achieved successful quantization of 4/8- bits for these models.

#### 2) COLLABORATIVE REASONING

Collaborative reasoning utilizes the cloud with higher reasoning performance as the reasoning backend to enhance inference. For reasoning, it directly on the edge side can have smaller latency and greater throughput, while directly on the cloud side can bring better reasoning accuracy. How to make the latency and throughput not significantly reduced and improve the reasoning accuracy with limited resources for edge-side reasoning has become an important research direction. Shao *et al.* [111] proposed a three-step inference framework and an incremental network pruning method. The former is used for communication-computation tradeoffs due to local computing load and communication overhead, while the latter is used to reduce redundant weights and computational delays. Simulation results show that the authors' proposed framework and method achieve the above objectives. Yang *et al.* [112] proposed a joint inference task selection and downlink beamforming strategy and a method called GSBF with the aim of minimizing the total power consumption and improving energy efficiency. Simulation results verify that the scheme improves the competitive performance of edge AI inference systems. Wang *et al.* [113] proposed the use of DRL to optimize edge caching and computation, and to further better deploy resource management, a framework called ''in - edge AI'' was investigated. Simulation results show that the scheme can improve the balance of performance and cost. Yang *et al.* [114] proposed a communication-efficient edge inference design and a low-latency data shuffling strategy. In edge AI inference, the authors' scheme excels in terms of latency and energy efficiency. Li *et al.* [115] studied AI service provisioning at the edge of a 6G network and proposed a resource pooling method to achieve data management and resource consumption for network slicing. In addition, the method can determine the training location and training method of AI models based on the data availability, resource constraints, and business performance requirements in the network. Yang *et al.* [116] proposed a decentralized model learning framework called E-Tree and a KMA algorithm based on this framework. The results show that E-Tree outperforms other model learning methods (e.g., Joint Learning and Gossip Learning). Wan *et al.* [117] investigated a chip called RRAM that provides a high degree of generality for different model architectures through collaborative optimization at all design levels, from algorithms and architectures to circuits and devices. This work provides lessons for building efficient and reconfigurable edge AI hardware platforms. Long *et al.* [118] proposed an architecture called MEANet for distributed training and inference between the edge and the cloud. The simulation results show that the model proposed by the authors outperforms the standard model in terms of accuracy and energy consumption.

#### 3) FRAME DESIGNS

The development of a hardware and software framework for handling EI computing is a key problem to be addressed. Du *et al.* [119] proposed a new hierarchical stochastic gradient quantization framework and investigated its impact on the learning performance, which reduces the communication overhead. In addition, the framework's bit allocation scheme reduces quantization errors. By testing, the framework greatly reduces the communication overhead with similar guaranteed accuracy compared to the state-of-the-art signSGD scheme. Khoram *et al.* [120] proposed a framework called TOCO to address the limitations of different edge devices deployed in large models. It uses an in-depth analysis of the model to maintain accuracy, and the analysis results in tolerances that can be used to perform compression in a fine-grained manner. Li *et al.* [121] designed a framework, Edgent, which utilizes edge computing for collaborative DNN inference through things-edge collaboration. Since dividing and resizing the DNNs, the inference accuracy is improved and the computational latency is reduced. The evaluation results validate the effectiveness of the framework. Liu *et al.* [122] proposed a framework, HierTrain, and a new hybrid parallel approach. The former can efficiently deploy DNN training tasks into a hierarchical MECC architecture, and the latter can adaptively distribute DNN model layers and data samples across three layers. Simulation results show that HierTrain can achieve a speedup of 6.9x compared to the cloud-based hierarchical training method.

Wang *et al.* [123] introduced KubeEdge, a kubernetes-based edge computing framework. It provides resource management, deployment, operation and synergy for edge computing. Rexha *et al.* [124] proposed an edge/cloud-based telemetry framework that can collect relevant data and transmitting it to a cloud-based system for processing and receiving feedback operations. The results show that the framework is capable of efficiently executing edge AI applications. Gerlinghoff *et al.* [125] proposed an E2E framework,

E3NE, that automatically generates efficient SNN inference logic for FPGA and enhances scalability and generality. The results show that the framework can reduce energy consumption and latency on top of saving hardware resources.

#### 4) HARDWARE ACCELERATION

AI is rapidly moving from data centers to edge computing, and developers typically use general-purpose CPU and GPU cores to develop and train neural network models, but these cores are far less efficient than dedicated accelerators for inference tasks. Liang *et al.* [126] compared the advantages and limitations of a dedicated edge system using edge accelerators with more traditional forms of edge and cloud computing. The results show that the former can provide better performance in terms of power and cost. Hao *et al.* [127] provided an effective solution for three algorithm/accelerator co-design methods. The effectiveness of the co-design approaches is demonstrated through extensive experiments on FPGAs and GPUs. Liang *et al.* [128] designed an analytical model for DNN inference work on a shared edge accelerator. The algorithm designed by using this model is able to manage multiple applications on the edge accelerators intelligently. Simulation results show that the scheme is able to predict latency behavior and improve resource sharing efficiency.

### C. EDGE COMPUTING COMBINED WITH BLOCKCHAIN

Edge computing, closed to the data source side, is a comprehensive platform that provides the integrate of network, computing, storage and application core functions. Meanwhile, blockchain is essentially a new application model based on the combination of distributed data storage, peer-to-peer transmission, consensus mechanisms, cryptographic algorithms and other computer technologies, therefore a feasibility study is conducted on edge computing combined with blockchain technology.

#### 1) THE PRINCIPLE OF BLOCKCHAIN

Blockchain first appeared in the concept of Bitcoin proposed by Satoshi Nakamoto in 2008. Blockchain, as Bitcoin bookkeeping technology, is not a separate technology, but a new application model of multiple computer technologies. Blockchain is commonly known as the storage of data, but it has the following of three main features compared to the usual database: 1) the data is open and transparent; 2) the history of the data is traceable; 3) the data cannot be tampered with. Blockchain applications have now been extended to digital asset management, IoT, smart manufacturing, supply chain finance and many other fields. The chain storage structure of blockchain consists of individual blocks, each of which is connected to the previous block through a hash tag in the block header, thus forming a one-way chain structure, with the first block being called the founding block. Each block contains two parts: block header and block body, where the block header contains 80 B keyword identification, and the block body mainly contains transaction information and other data. Blockchain mainly contains the following key technologies: 1) distributed ledger; 2) consensus mechanism; 3) cryptographic features; and 4) smart contracts.

#### 2) INTEGRATION NEEDS

The distributed characteristics of edge computing in computing, storage, and networking coincide with the decentralized model of blockchain, and the service focus is all geared toward enterprise and vertical application industries. It has the main integration needs, as following: 1) blockchain nodes can be deployed on edge nodes which has the computing ability to provide resources, communication and capacity for edge computing services; 2) blockchain provides a secure and trustworthy environment for edge computing, in order to ensure the integrity and authenticity of data storage; 3) blockchain combined with edge computing can form an efficient platform of information and value to promote sharing resource and optimal allocation.

#### 3) DEPLOYMENT MODES

Blockchain combined with edge computing will become an important network infrastructure and innovation driver for operators in the 5G era, and its deployment model is studied and discussed below. Liao *et al.* [129] designed a blockchain and smart contract-based scheme for secure task offloading and a framework called QUOTA-UCB. The authors verified the reliability, feasibility and effectiveness of the proposed scheme through extensive theoretical analysis and simulations. Li *et al.* [130] introduced techniques such as UAV, blockchain and MEC, and proposed a joint optimization framework, which was formulated as MDP. To solve dynamic and complex optimization problems, dueling DQN was used for optimization selection and decision making. Simulation results show that the proposed framework can effectively improve the system throughput and revenue. Liu *et al.* [131] proposed a MECO-enabled transcoding framework for blockchain-based video streams, and then used the ADMM method to solve the video transcoding and block size problems and used smart contracts to achieve distributed optimization among untrustworthy entities. Simulation results verify the effectiveness of the framework. Feng *et al.* [132] proposed a joint optimization framework for blockchain-enabled MEC systems. The equilibrium requirement between energy consumption and DTF is modeled as a MINLP problem, and then the optimization variables are decoupled for this problem to achieve efficient algorithm design. Simulation results show that the scheme is able to achieve a balance between performance. Chu *et al.* [133] proposed a scalable blockchain and a neural network-based task offloading technique for MEC scenarios. The approach has good scalability in mobile scenarios. Guo *et al.* [134] proposed a framework called B-MEC to address the throughput and user QoS of this system. in addition, double-dueling DQN was utilized to cope with its dynamic nature. Simulation results show the effectiveness of the approach. Dai *et al.* [135] proposed a

blockchain-empowered distributed content caching framework to formulate the content caching problem in the form of DRL and design a new DRL-based content caching scheme to achieve maximum content caching and cope with vehicle mobility. Numerical results demonstrate the effectiveness of the scheme. Zhang *et al.* [136] proposed a blockchain-based message transmission mechanism and a credit mechanism. Simulation results show that the scheme can improve the reliability of data transmission. Liu *et al.* [137] proposed a blockchain authorized vehicle group authentication scheme based on secret sharing and dynamic agent mechanism to achieve collaborative authentication and decentralized authentication. The results show that the scheme not only minimizes the communication overhead and computation, but also achieves collaborative vehicle privacy protection. Guo *et al.* [138] constructed a network called CMN to cope with the computational cost of mobile devices in the blockchain mining process. On the one hand, the BNE method is applied to solve the optimal auction price. On the other hand, the Stackelberg game obtains the optimal resource price and the demand for device resources. The simulation results show that the mechanism maximizes the profit and the increase of utility in the mining network. Gao *et al.* [139] designed a framework called B-ReST, which defines the physical architecture, functional architecture, and workflow. In addition, a DRL-based approach was used to solve the RPM problem. Simulation results show that the framework improves the capabilities in terms of resource sharing and transaction processing. Gupta *et al.* [140] proposed a blockchain-based EI system that addresses the security, privacy, latency and efficiency of CED data. Zhang *et al.* [141] proposed a secure mobility management framework for ultra-dense edge computing based on blockchain. In addition, the wireless switching and service migration decisions between base stations were transformed into a multi-objective dynamic optimization problem using Lyapunov optimization, and then the optimization problem was solved using the DRL method. The results show that the scheme outperforms existing schemes in terms of average delay, task failure rate, and switching rate of computational tasks. Rivera *et al.* [142] proposed a blockchain framework for providing a trusted collaboration mechanism between edge servers in a MEC environment, and experimentally evaluated the scheme using the Caliper tool and Hyperledger Fabric benchmarks. Islam *et al.* [143] proposed a decentralized blockchain-based architecture and a secure IVEC federation model, which improves the transparency and balances the load of IVEC resource management. Gumaei *et al.* [144] introduced a framework that combines blockchain with DRNN and edge computing for 5G UAV identification and flight pattern detection. The scheme is higher in detection accuracy than other existing DL models. Li *et al.* [145] proposed a three-layer network model called BMEC. Firstly, the cloned blocks are identified by the NCBI method. Secondly, the blockchain network is divided using the Prim algorithm. The experimental results show that the blockchain construction latency of this scheme is smaller compared to the traditional edge computing methods. Zhang *et al.* [146] proposed an architecture, LBC, and an attribute-based cryptographic access control scheme, ABE-ACS. simulation results show that the scheme improves throughput and reduces energy consumption while ensuring privacy security. Nguyen *et al.* [147] proposed an architecture, BFL, and a series of solutions which include offloading strategies, ML model aggregation and a new DRL approach. Simulation results show that these measures outperformed existing methods in terms of training efficiency, convergence speed and latency.

In summary: Academic research on edge computing focuses on two key technology directions: first, edge-native technologies directly related to edge computing, currently represented by resource scheduling and EI; second, the convergence of edge computing with various ICT frontier technologies, such as the combination of edge computing and blockchain.

## V. THE DEVELOPMENT PATHS AND TRENDS OF EDGE COMPUTING

The edge computing industry is an ecosystem consisting of multiple communities of interest such as telecom operators, telecom equipment vendors, IT vendors, third-party application developers, content providers, and users.

### A. 5G NATIVELY SUPPORTS EDGE COMPUTING CAPABILITIES

Since the global 5G network construction in 2019 and the announcement of 5G commercialization by dozens of mainstream operators in 2020, it has been introduced to multiple vertical industries such as Industrial Internet, autonomous drive, smart cities, and smart factories. MEC enables operators to divert service at the edge of the network, and various edge computing service/product providers choose different entry points based on their own advantages and application scenario characteristics. Currently, the three major domestic operators are actively introducing MEC capabilities to vertical industries to enhance network value.

The current 5G MEC route is as follows: 1) since 2014, ETSI has extended the concept of edge computing to multi-access edge computing, focusing on new services and needs such as 5G, Wi-Fi, etc.; 2) since 2017, 3GPP has been leading the development of relevant standards with the industry to ensure the completeness of 5G standards, support MEC and ensure subsequent enhancements; 3) also since 2017, CCSA has carried out MEC standardization work and has developed nearly 10 standards, mainly exploring the standardization of MEC platform technical requirements, capability opening, and security technical requirements [148].

#### 1) 5G MEC DEVELOPMENT FACES BOTTLENECKS AND FUTURE TRENDS

5G MEC has already started pilot applications in multiple industries, but the overall development is still at an early

stage, and there are some problems and challenges: 1) the infrastructure construction model needs to be explored, and currently the construction of MEC nodes is mainly undertaken independently by telecom operators, which leads to high costs of building them. This may be difficult to cope with the future large-scale deployment of MEC construction; 2) the application ecology is not yet mature, although all parties are currently exploring application models for vertical industries, such as MEC networks to interconnect with the enterprise intranet of vertical industries and integrate 5G communication capabilities and MEC applications into business systems, the relevant requirements have not been standardized and the application ecology is still not established.

Reference can be made to the strategies of foreign operators in this regard, for example, AT&T launched Akraino, an open-source platform for edge computing, to establish a strategic foundation, and Deutsche Telekom set up MobiledgeX, a subsidiary specializing in edge computing services and products, to develop a cross operator mobile edge computing platform. South Korea's SKT has launched an open platform for edge computing to create 5G and MEC ecosystem connecting developers and enterprise users.

### B. CLOUD-NATIVE EDGE COMPUTING DEVELOPMENT PATHS

Cloud-native powering edge computing brings solutions to the increasingly complex management of edge environments. Several frameworks for edge computing projects have emerged in the industry and academia, such as OpenYurt, a project open sourced by Aliyun, KubeEdge by Huawei, and EdgeX Foundry operated by the Linux Foundation.

OpenYurt is a framework built on top of Kubernetes that overcomes some of the limitations of edge scenarios, such as how to minimize long-distance network traffic between devices and workloads, how to address reliability in edge scenarios, how to perform secure authentication, how to reduce transport latency, etc. OpenYurt provides full Kubernetes API compatibility and supports all features of Kubernetes, which also provides a tool to convert native Kubernetes to edge state and also improves the stability of the cluster in edge scenarios [149].

KubeEdge is based on the Kubernetes architecture and provides functional support for many edge scenarios, unifying development, deployment and management views, which enhances offline operational capabilities, edge-cloud collaboration capabilities and edge collaboration capabilities. The architecture uses cloud components and edge components, which serve the following purposes: 1) in the cloud component, users issue commands to the expected state of the target object via the kubectl command line, which is received by the Kubernetes API server and dispatched to the object using the scheduler; 2) in the edge component, the design principle is based on simplicity to reduce the resource footprint, the probability of failure, and the difficulty of maintenance of the edge component [150].

EdgeX Foundry is positioned as a generic framework for general-purpose industrial IoT edge computing, deployed on edge devices such as routers and switches to provide plug-and-play functionality to various sensors, devices or other IoT devices. In addition, it will collect and analyze this generated data exporting it to edge computing applications or cloud computing centers for further processing [151].

As an extension and supplement of cloud computing, edge computing has formed a consensus in the field of cloud computing. Therefore, IT companies hope to extend cloud computing capabilities with cloud-native edge computing to protect their core competitive advantages in the original domain and form an integrated synergy of cloud, edge and terminal.

#### 1) THE DEVELOPMENT OF CLOUD-NATIVE EDGE COMPUTING FACES BOTTLENECKS AND FUTURE TRENDS

Cloud-native edge computing is essentially a combination of traditional cloud computing technologies lightened and then combined with new edge-native technologies to achieve fast response and scalability of computing, storage, network and other resources. Although cloud-native has great potential to drive the development of edge computing, and industry has launched related solutions one after another, it is still in the initial stage of research in this direction. Due to the essential difference between edge computing environment and cloud computing data center, there are still many challenges in cloud-native edge computing: 1) as an emerging technology concept, edge computing has not yet fully matured, and there is confusion in the industry about the understanding of cloud-native edge computing and MEC concepts, and the business relationship between MEC platforms of operators and edge-cloud platforms of IT enterprises is still unclear; 2) edge-cloud collaboration is an important architecture for IT vendors to drive the development of central cloud to the edge side. However, in the process of implement, there are still problems such as lack of unified application management northbound interface, difficulties in application as well as service distribution, and lack of application distribution mechanism across edge clouds.

Currently, IT vendors are actively promoting edge-cloud collaboration practices, propelling cloud computing services and cloud-native capabilities from multiple dimensions such as resources, data and applications, and accelerating the construction of cloud-native edge computing digital transformation solutions.

In summary: The edge computing industry is driven by different subjects, and two major development paths have been formed: one is the vertical industry path of 5G MEC development led by telecom operators; the other is the development path of cloud-native edge computing led by IT enterprises.

## VI. CHALLENGES AND KEY ISSUES

Although a lot of results have been accumulated from related research in edge computing, there are still many key issues

that have not been well explored. This section discusses several open challenges and key questions for future research.

### A. MODEL AND ARCHITECTURE

#### 1) COMPUTATION AND COMMUNICATION MODEL

The two most important decision objectives of the computational model are time and energy, respectively. In other words, minimizing the delay time and energy consumption is the goal of the optimal solution in the optimization process. In order to efficiently realize the task processing of edge resources, a computing model needs to be established to reflect the relationship between task data size and computing power. In most existing works, the computing power required for task processing is directly proportional to the product of task data size and processing density [14], [15], [79]. However, due to the different types of tasks in the environment of edges, there will be different processing density. Therefore, more flexible calculation models need to be further studied. additionally, because the channel conditions of the real edge environments are often unstable, it is necessary to develop communication models suitable for different scenarios through field tests.

#### 2) COMPUTATION MIGRATION

Computation migration is extremely important due to the capacity, resource, bandwidth, energy, and other constraints of edge nodes. However, computation migration itself is a complex process, and in most existing works considering computation migration, only the migration decision step is considered, while other steps (task upload, MEC server execution, result return, etc.) are ignored [60], [61], [64]. In the current research, most migration strategies only consider computation latency or terminal energy consumption, and the global optimization of both has not been achieved. Future research can pay more attention to the new model of migration strategy and the reduction of computing complexity.

#### 3) HETEROGENEOUS ARCHITECTURE

From our survey, we know that the current architecture of edge computing usually consists of a thing layer, an edge layer and a cloud layer. However, air-ground cooperative MEC will be the trend to provide high-quality intelligent services for future 6G networks, but heterogeneous nodes will make the management and scheduling of resources more challenging [152]. Therefore, it is necessary to develop a technology for effective resource scheduling and management, such as network slicing that enables dynamic and efficient network management of resources in heterogeneous nodes.

### B. EI'S KEY OPEN CHALLENGES

#### 1) PROGRAMMING AND SOFTWARE PLATFORMS

The potential of EI services can be realized by programming/software platforms to provide edge computing services. However, most of these platforms are currently used to connect powerful cloud data centers and do not fully exploit the benefits of edge computing. With the emergence of computationally intensive IoT applications, the need for EI services has become more urgent.

#### 2) EI MODEL DEPLOYMENT

At present, model training optimization techniques for EI are mainly divided into five types, including federal learning, parameter aggregation optimization, gradient compression, model partitioning and migration learning. With the further integration of edge computing, cloud computing and high-performance computing, EI and cloud intelligence will be an important cornerstone to support AI applications. Looking ahead, edge-cloud intelligence collaborative architecture, arithmetic-aware interconnection, automatic model design, and distributed sharing incentive mechanism will be important research directions.

#### 3) COMPUTATION-AWARE NETWORK TECHNOLOGIES

For EI, applications based on computing intensive AI usually run-in distributed edge computing environment. Therefore, Computation-intensive AI applications can be achieved by integrating the functions of 5G, URLLC, SDN, and NFC with edge computing to provide ultra-reliable, low-latency services. Through these technologies, flexible control of network resources will be realized to support on-demand interconnection across different edge nodes. On the other hand, computing aware communication technology has also begun to attract people's attention, such as gradient coding to reduce the spurious effect in distributed learning, and air computing for distributed intelligent learning, which are useful for the training acceleration of EI model.

### C. BEOT'S KEY OPEN CHALLENGES

#### 1) SECURITY AND PRIVACY

The integration of edge computing and blockchain can improve the overall performance of IoT devices. Taking the IoT devices group as an example, on the one hand, MEC can act as the ''local brain'' of IoT devices, storing and processing the data returned from different IoT devices in the same scene, and optimizing and correcting the working state and path of various devices to achieve the optimal overall application of the scene. On the other hand, terminal equipment can ''host'' data to the edge computing server and ensure the reliability and security of the data with the help of blockchain technology, but, also, for the future IoT devices according to service charges and other development methods provide the possibility. Nevertheless, the edge outsourcing services of integrated blockchain and edge computing systems present new security and privacy challenges [31]. The most used sidechain solutions in existing works may experience transaction losses in the extreme case of node channel crash. In the future, with the breakthrough in the application of blockchain technology, the security problems faced by edge devices will

be solved and the decentralization of the IoE will be truly realized.

#### 2) FUNCTION INTEGRATION

BEoT integrates multiple platforms, network architectures and servers. Therefore, it is difficult to unify the management of storage servers running on different operating systems. An alternative solution to the inefficiency and high cost of blockchain storage is proposed, which is to use IPFS to store file data and put permanently available unique IPFS addresses into blockchain transactions without putting the data itself into the blockchain. However, this requires consideration of integration flexibility and stability.

#### 3) RESOURCE MANAGEMENT

Under the edge computing based on blockchain, the cooperation between servers is more frequent and the scope of resource sharing is more extensive. Such as the large-scale optimization of edge server cooperation and the management of dynamic resources, these problems are more serious in the integration of blockchain and edge computing. How to design a multi-criteria scheduler based on blockchain to achieve multi-functional joint optimization is a challenge. In addition, the resources consumed by blockchain can not be ignored, so the edge computing resource management of proof of work also needs to be actively explored.

## VII. CONCLUSION

In this survey, we conduct a systematic and comprehensive review of the development of edge computing. First, we provide a brief review of the latest edge computing literature. Second, we find that there is a lack of research on the overall overview of the latest developments in edge computing. Third, to fill this gap, we then provide an in-depth overview of the latest technologies in edge computing, especially from the perspective of architectures and models, key technologies, and directions, which are the outstanding results of this survey. Regarding the key research questions, we first summarize various architectures and models of edge computing, which include generic architectures and models applied under relevant popular domains. In terms of key technologies, we investigate and clearly classify three research aspects, namely, resource scheduling, EI, and edge computing combined with blockchain. In addition, the development paths and trends of edge computing industry are summarized. Finally, we clarify the current research challenges and key issues, and expect to convey ideas and solutions that can improve the development of edge computing and help people better understand edge computing at a higher level. We believe that our research provides timely guidance to researchers, engineers, educators, and readers on the latest developments in edge computing.

## REFERENCES


[1] B. Rajkumar and N. S. Satish, "Management and orchestration of network slices in 5G, fog, edge, and clouds," *Fog Edge Comput., Princ. Paradigms*, vol. 8, pp. 79–101, Jan. 2019.

[2] S. D. A. Shah, M. A. Gregory, and S. Li, "Cloud-native network slicing using software defined networking based multi-access edge computing: A survey," *IEEE Access*, vol. 9, pp. 10903–10924, 2021, doi: 10.1109/ACCESS.2021.3050155.

[3] J. Pan, Y. Liu, J. Wang, and A. Hester, "Key enabling technologies for secure and scalable future fog-IoT architecture: A survey," 2018, *arXiv:1806.06188*.

[4] K. S. Kumar, A. S. Radhamani, and S. Sundaresan, "Proficient approaches for scalability and security in IoT through edge/fog/cloud computing: A survey," *Int. J. Data. Sci.*, vol. 6, no. 1, pp. 33–44, Aug. 2021, doi: 10.1504/IJDS.2021.117465.

[5] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge-computing-assisted Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4004–4022, Mar. 2021, doi: 10.1109/JIOT.2020.3015432.

[6] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018, doi: 10.1109/JIOT.2017.2750180.

[7] A. Hamm, A. Willner, and I. Schieferdecker, "Edge computing: A comprehensive survey of current initiatives and a roadmap for a sustainable edge computing development," 2019, *arXiv:1912.08530*.

[8] N. C. Luong, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource management in cloud networking using economic analysis and pricing models: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 954–1001, 2nd Quart., 2017, doi: 10.1109/COMST.2017.2647981.

[9] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: A survey on architectures, infrastructure, and algorithms," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–37, Oct. 2019, doi: 10.1145/3326066.

[10] M. Ghobaei-Arani, A. Souri, and A. A. Rahmanian, "Resource management approaches in fog computing: A comprehensive review," *J. Grid Comput.*, vol. 18, no. 1, pp. 1–42, Mar. 2020, doi: 10.1007/s10723-019-09491-1.

[11] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, Sep. 2017, doi: 10.1109/COMST.2017.2682318.

[12] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, "Edge cloud offloading algorithms: Issues, methods, and perspectives," *ACM Comput. Surveys*, vol. 52, no. 1, pp. 1–23, Jan. 2020, doi: 10.1145/3284387.

[13] R. A. Dziyauddin, D. Niyato, N. C. Luong, M. A. M. Izhar, M. Hadhari, and S. Daud, "Computation offloading and content caching delivery in vehicular edge computing: A survey," 2019, *arXiv:1912.07803*.

[14] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "A survey and taxonomy on task offloading for edge-cloud computing," *IEEE Access*, vol. 8, pp. 186080–186101, 2020, doi: 10.1109/ACCESS.2020.3029649.

[15] M. Algarni, A. Cherif, and E. Alkayal, "A survey of computational offloading in cloud/edge-based architectures: Strategies, optimization models and challenges," *KSII. Trans. Internet. Inf. Syst.*, vol. 15, pp. 952–973, Oct. 2021, doi: 10.3837/tiis.2021.03.008.

[16] A. Islam, A. Debnath, M. Ghose, and S. Chakraborty, "A survey on task offloading in multi-access edge computing," *J. Syst. Archit.*, vol. 118, Sep. 2021, Art. no. 102225, doi: 10.1016/j.sysarc.2021.102225.

[17] T. L. Duc, R. G. Leiva, P. Casari, and P.-O. Östberg, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey," *ACM Comput. Surveys*, vol. 52, no. 5, pp. 1–39, Sep. 2020, doi: 10.1145/3341145.

[18] F. Spinelli and V. Mancuso, "Toward enabled industrial verticals in 5G: A survey on MEC-based approaches to provisioning and flexibility," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 596–630, Nov. 2021, doi: 10.1109/COMST.2020.3037674.

[19] D. Wen, X. Li, Q. Zeng, J. Ren, and K. Huang, "An overview of data-importance aware radio resource management for edge machine learning," *J. Commun. Inf. Netw.*, vol. 4, no. 4, pp. 1–14, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9005429, doi: 10.23919/JCIN.2019.9005429.

[20] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: A survey," *ACM Comput. Surveys*, vol. 54, no. 8, pp. 1–37, Nov. 2022, doi: 10.1145/3469029.

[21] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107496, doi: 10.1016/j.comnet.2020.107496.

[22] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, and E. Alanazi, "Applying machine learning techniques for caching in next-generation edge networks: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 181, May 2021, Art. no. 103005, doi: 10.1016/j.jnca.2021.103005.

[23] S. Voghoei, N. H. Tonekaboni, J. G. Wallace, and H. R. Arabnia, "Deep learning at the edge," in *Proc. Int. Conf. CSCI*, Las Vegas, NV, USA, 2018, pp. 895–901.

[24] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, Jan. 2020, doi: 10.1109/COMST.2020.2970550.

[25] Y. Sun, H. Ochiai, and H. Esaki, "Decentralized deep learning for multi-access edge computing: A survey on communication efficiency and trustworthiness," 2021, *arXiv:2108.03980*.

[26] H. Qiu, K. Zhu, N. C. Luong, C. Yi, D. Niyato, and D. I. Kim, "Applications of auction and mechanism design in edge computing: A survey," 2021, *arXiv:2105.03559*.

[27] J. Moura and D. Hutchison, "Game theory for multi-access edge computing: Survey, use cases, and future trends," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 260–288, Aug. 2019, doi: 10.1109/COMST.2018.2863030.

[28] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017, doi: 10.1109/COMST.2017.2705720.

[29] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020, doi: 10.1109/ACCESS.2020.3001277.

[30] A. Al-Ansi, A. M. Al-Ansi, A. Muthanna, I. A. Elgendy, and A. Koucheryavy, "Survey on intelligence edge computing in 6G: Characteristics, challenges, potential use cases, and market drivers," *Future Internet*, vol. 13, no. 5, p. 118, Apr. 2021, doi: 10.3390/fi13050118.

[31] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, Feb. 2019, doi: 10.1109/COMST.2019.2894727.

[32] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5G and beyond networks: A state of the art survey," *J. Netw. Comput. Appl.*, vol. 166, Sep. 2020, Art. no. 102693, doi: 10.1016/j.jnca.2020.102693.

[33] A. Queiroz, E. Oliveira, M. Barbosa, and K. Dias, "A survey on blockchain and edge computing applied to the internet of vehicles," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, New Delhi, India, Dec. 2020, pp. 1–6.

[34] T. R. Gadekallu, Q.-V. Pham, D. C. Nguyen, P. K. R. Maddikunta, N. Deepa, B. Prabadevi, P. N. Pathirana, J. Zhao, and W.-J. Hwang, "Blockchain for edge of things: Applications, opportunities, and challenges," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 964–988, Jan. 2022, doi: 10.1109/JIOT.2021.3119639.

[35] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-computing-enabled smart cities: A comprehensive survey," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10200–10232, Oct. 2020, doi: 10.1109/JIOT.2020.2987070.

[36] P. Boccadoro, "Smart grids empowerment with edge computing: An overview," 2018, *arXiv:1809.10060*.

[37] Z. Li, X. Zhou, and Y. Qin, "A survey of mobile edge computing in the industrial internet," in *Proc. 7th Int. Conf. Inf., Commun. Netw. (ICICN)*, Apr. 2019, pp. 94–98.

[38] A. Sufian, A. Ghosh, A. S. Sadiq, and F. Smarandache, "A survey on deep transfer learning to edge computing for mitigating the COVID-19 pandemic," *J. Syst. Archit.*, vol. 108, Sep. 2020, Art. no. 101830, doi: 10.1016/j.sysarc.2020.101830.

[39] G. M. Bianco, C. Occhiuzzi, N. Panunzio, and G. Marrocco, "A survey on radio frequency identification as a scalable technology to face pandemics," *IEEE J. Radio Freq. Identificat.*, vol. 6, pp. 77–96, 2022, doi: 10.1109/JRFID.2021.3117764.

[40] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 140–146, Feb. 2020, doi: 10.1109/MWC.001.1800594.

[41] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2020, doi: 10.1109/JPROC.2019.2947490.

[42] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, vol. 26, no. 3, pp. 1145–1168, 2021, doi: 10.1007/s11036-020-01624-1.

[43] D. R. Patrikar and M. R. Parate, "Anomaly detection using edge computing in video surveillance system: Review," 2021, *arXiv:2107.02778*.

[44] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A survey on edge computing systems and tools," *Proc. IEEE*, vol. 107, no. 8, pp. 1537–1562, Aug. 2019, doi: 10.1109/JPROC.2019.2920341.

[45] L. T. Van, N. E. Ioini, C. Pahl, and H. R. Barzegar, "Edge computing simulation platforms: A technology survey," in *Proc. Eur. Conf. (SOCC)*. Cham, Switzerland: Springer, 2020, pp. 18–28.

[46] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–8.

[47] J. Ren, Y. He, G. Huang, G. Yu, Y. Cai, and Z. Zhang, "An edge-computing based architecture for mobile augmented reality," *IEEE Netw.*, vol. 33, no. 4, pp. 162–169, Jul. 2019, doi: 10.1109/MNET.2018.1800132.

[48] A. Willner and V. Gowtham, "Toward a reference architecture model for industrial edge computing," *IEEE Commun. Standards Mag.*, vol. 4, no. 4, pp. 42–48, Dec. 2020, doi: 10.1109/MCOMSTD.001.2000007.

[49] H. Rahimi, Y. Picaud, K. D. Singh, G. Madhusudan, S. Costanzo, and O. Boissier, "Design and simulation of a hybrid architecture for edge computing in 5G and beyond," *IEEE Trans. Comput.*, vol. 70, no. 8, pp. 1213–1224, Aug. 2021, doi: 10.1109/TC.2021.3066579.

[50] X. Wang, J. Ye, and J. C. S. Lui, "Decentralized task offloading in edge computing: A multi-user multi-armed bandit approach," 2021, *arXiv:2112.11818*.

[51] L. Ale, S. A. King, N. Zhang, A. R. Sattar, and J. Skandaraniyam, "D3PG: Dirichlet DDPG for task partitioning and offloading with constrained hybrid action space in mobile edge computing," 2021, *arXiv:2112.09328*.

[52] Z. Liang, H. Chen, Y. Liu, and F. Chen, "Data sensing and offloading in edge computing networks: TDMA or NOMA?" *IEEE Trans. Wireless Commun.*, early access, Dec. 1, 2021, doi: 10.1109/TWC.2021.3130599.

[53] Y. Su, W. Fan, Y. Liu, and F. Wu, "Game-based pricing and task offloading in mobile edge computing enabled edge-cloud systems," 2021, *arXiv:2101.05628*.

[54] Z. Li, X. Zhou, Y. Liu, C. Fan, and W. Wang, "A computation offloading model over collaborative cloud-edge networks with optimal transport theory," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Guangzhou, China, Dec. 2020, pp. 1006–1011.

[55] G. Qu, H. Wu, R. Li, and P. Jiao, "DMRO: A deep meta reinforcement learning-based task offloading framework for edge-cloud computing," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 3, pp. 3448–3459, Sep. 2021, doi: 10.1109/TNSM.2021.3087258.

[56] M. Ke, Z. Gao, Y. Wu, X. Gao, and K.-K. Wong, "Massive access in cell-free massive MIMO-based Internet of Things: Cloud computing and edge computing paradigms," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 756–772, Mar. 2021, doi: 10.1109/JSAC.2020.3018807.

[57] S. Tuli, G. Casale, and N. Jennings, "MCDS: AI augmented workflow scheduling in mobile edge cloud computing systems," *IEEE Trans. Parallel Distrib. Syst.*, early access, Dec. 16, 2021, doi: 10.1109/TPDS.2021.3135907.

[58] J. He, Y. Wang, X. Du, and Z. Lu, "V2V-based task offloading and resource allocation in vehicular edge computing networks," 2021, *arXiv:2112.15065*.

[59] Y. Miao, G. Wu, M. Li, A. Ghoneim, M. Al-Rakhami, and M. S. Hossain, "Intelligent task prediction and computation offloading based on mobile-edge cloud computing," *Future Gener. Comput. Syst.*, vol. 102, pp. 925–931, Jan. 2020, doi: 10.1016/j.future.2019.09.035.

[60] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu, and I. Humar, "A dynamic service migration mechanism in edge cognitive computing," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–15, May 2019, doi: 10.1145/3239565.

[61] C. Chang, A. Hadachi, and S. Srirama, "Adaptive edge process migration for IoT in heterogeneous cloud-fog-edge computing environment," 2018, *arXiv:1811.10939*.

[62] A. Yousafzai, I. Yaqoob, M. Imran, A. Gani, and R. M. Noor, "Process migration-based computational offloading framework for IoT-supported mobile edge/cloud computing," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4171–4182, May 2020, doi: 10.1109/JIOT.2019.2943176.

[63] M. V. Ngo, T. Luo, H. T. Hoang, and T. Q. S. Ouek, "Coordinated container migration and base station handover in mobile edge computing," in *Proc. IEEE Global Commun. Conf.*, Taipei, Taiwan, Dec. 2020, pp. 1–6.

[64] J. Wang, J. Hu, and G. Min, "Online service migration in edge computing with incomplete information: A deep recurrent actor-critic method," 2020, *arXiv:2012.08679*.

[65] Z. Liang, Y. Liu, T.-M. Lok, and K. Huang, "Multi-cell mobile edge computing: Joint service migration and resource allocation," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5898–5912, Sep. 2021, doi: 10.1109/TWC.2021.3070974.

[66] M. Xu, Q. Zhou, H. Wu, W. Lin, K. Ye, and C. Xu, "PDMA: Probabilistic service migration approach for delay-aware and mobility-aware mobile edge computing," *Softw., Pract. Exper.*, vol. 52, no. 2, pp. 394–414, Feb. 2022, doi: 10.1002/spe.3014.

[67] M. Goudarzi, M. Palaniswami, and R. Buyya, "A distributed application placement and migration management techniques for edge and fog computing environments," in *Proc. Ann. Comput. Sci. Inf. Syst.*, Sofia, Bulgaria, Sep. 2021, pp. 37–56.

[68] T. He, A. N. Toosi, and R. Buyya, "Efficient large-scale multiple migration planning and scheduling in SDN-enabled edge computing," 2021, *arXiv:2111.08936*.

[69] K. Ha, Y. Abe, T. Eiszler, Z. Chen, W. Hu, B. Amos, R. Upadhyaya, P. Pillai, and M. Satyanarayanan, "You can teach elephants to dance: Agile VM handoff for edge computing," in *Proc. 2nd ACM/IEEE Symp. Edge Comput.*, San Jose, CA, USA, Oct. 2017, pp. 1–14.

[70] P. Mahadevappa and R. K. Murugesan, "A data quarantine model to secure data in edge computing," 2021, *arXiv:2111.07672*.

[71] T. Dlamini and A. F. Gambin, "Adaptive resource management for a virtualized computing platform within edge computing," in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Boston, MA, USA, Jun. 2019, pp. 1–9.

[72] S. Wan, J. Lu, P. Fan, and K. B. Letaief, "Toward big data processing in IoT: Path planning and resource management of UAV base stations in mobile-edge computing system," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5995–6009, Jul. 2020, doi: 10.1109/JIOT.2019.2954825.

[73] S. Jošilo and G. Dán, "Joint wireless and edge computing resource management with dynamic network slice selection," 2020, *arXiv:2001.07964*.

[74] X. Chen, Z. Lu, W. Ni, X. Wang, F. Wang, S. Zhang, and S. Xu, "Cooling-aware resource allocation and load management for mobile edge computing systems," 2020, *arXiv:2006.10978*.

[75] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7947–7962, Dec. 2021, doi: 10.1109/TWC.2021.3088910.

[76] S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu, "When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2238–2251, Feb. 2021, doi: 10.1109/JIOT.2020.3026589.

[77] E. Moro and I. Filippini, "Joint management of compute and radio resources in mobile edge computing: A market equilibrium approach," *IEEE Trans. Mobile Comput.*, early access, Jun. 23, 2021, doi: 10.1109/TMC.2021.3091764.

[78] C. W. Zaw, S. R. Pandey, K. Kim, and C. S. Hong, "Energy-aware resource management for federated learning in multi-access edge computing systems," *IEEE Access*, vol. 9, pp. 34938–34950, 2021, doi: 10.1109/ACCESS.2021.3055523.

[79] J. Yan, S. Bi, Y. J. Zhang, and M. Tao, "Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 235–250, Jan. 2020, doi: 10.1109/TWC.2019.2943563.

[80] C. F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4132–4150, Feb. 2019, doi: 10.1109/TCOMM.2019.2898573.

[81] Y. Sun, X. Guo, J. Song, S. Zhou, Z. Jiang, X. Liu, and Z. Niu, "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3061–3074, Apr. 2019, doi: 10.1109/TVT.2019.2895593.

[82] W. Du, T. Lei, Q. He, W. Liu, Q. Lei, H. Zhao, and W. Wang, "Service capacity enhanced task offloading and resource allocation in multi-server edge computing environment," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Milan, Italy, Jul. 2019, pp. 83–90.

[83] F. Wang, J. Xu, and S. Cui, "Optimal energy allocation and task offloading policy for wireless powered mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2443–2459, Apr. 2020, doi: 10.1109/TWC.2020.2964765.

[84] S. Batewela, C.-F. Liu, M. Bennis, H. A. Suraweera, and C. S. Hong, "Risk-sensitive task fetching and offloading for vehicular edge computing," *IEEE Commun. Lett.*, vol. 24, no. 3, pp. 617–621, Mar. 2020, doi: 10.1109/LCOMM.2019.2960777.

[85] M. C. Lucic, H. Ghazzai, A. Alsharoa, and Y. Massoud, "A latency-aware task offloading in mobile edge computing network for distributed elevated LiDAR," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seville, Spain, Oct. 2020, pp. 1–5.

[86] M. Tang and V. W. S. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1985–1997, Jun. 2022, doi: 10.1109/TMC.2020.3036871.

[87] X. Huang, L. He, X. Chen, L. Wang, and F. Li, "Revenue and energy efficiency-driven delay constrained computing task offloading and resource allocation in a vehicular edge computing network: A deep reinforcement learning approach," *IEEE Internet Things J.*, early access, Sep. 29, 2021, doi: 10.1109/JIOT.2021.3116108.

[88] Z. Sun, "BARGAIN-MATCH: A game theoretical approach for resource allocation and task offloading in vehicular edge computing networks," 2022, *arXiv:2203.14064*.

[89] Y. Li, B. Yang, H. Wu, Q. Han, C. Chen, and X. Guan, "Joint offloading decision and resource allocation for vehicular fog-edge computing networks: A contract-Stackelberg approach," *IEEE Internet Things J.*, early access, Feb. 11, 2022, doi: 10.1109/JIOT.2022.3150955.

[90] A. Abouaomar, S. Cherkaoui, Z. Mlika, and A. Kobbane, "Resource provisioning in edge computing for latency-sensitive applications," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11088–11099, Jul. 2021, doi: 10.1109/JIOT.2021.3052082.

[91] F. D. Tilahun, A. T. Abebe, and C. G. Kang, "Multi-agent reinforcement learning for distributed joint communication and computing resource allocation over cell-free massive MIMO-enabled mobile edge computing network," 2021, *arXiv:2201.09057*.

[92] O. Ascigil, A. Tasiopoulos, T. K. Phan, V. Sourlas, I. Psaras, and G. Pavlou, "Resource provisioning and allocation in function-as-a-service edge-clouds," *IEEE Trans. Services Comput.*, early access, Jan. 18, 2021, doi: 10.1109/TSC.2021.3052139.

[93] M. Nasimi, M. A. Habibi, B. Han, and H. D. Schotten, "Edge-assisted congestion control mechanism for 5G network using software-defined networking," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Lisbon, Portugal, Aug. 2018, pp. 1–5.

[94] X. Gao, R. Liu, and A. Kaushik, "A distributed virtual network function placement approach in satellite edge and cloud computing," 2021, *arXiv:2104.02421*.

[95] M. H. Ly, T. Q. Dinh, and H. H. Kha, "Joint optimization of execution latency and energy consumption for mobile edge computing with data compression and task allocation," in *Proc. Int. Symp. Electr. Electron. Eng. (ISEE)*, Ho Chi Minh City, Vietnam, Oct. 2019, pp. 113–118.

[96] N. Li, J. Yan, Z. Zhang, J. F. Martinez, and X. Yuan, "Game theory based joint task offloading and resource allocation algorithm for mobile edge computing," in *Proc. 16th Int. Conf. Mobility, Sens. Netw. (MSN)*, Tokyo, Japan, Dec. 2020, pp. 791–796.

[97] X. Zhong, X. Wang, T. Yang, Y. Yang, Y. Qin, and X. Ma, "POTAM: A parallel optimal task allocation mechanism for large-scale delay sensitive mobile edge computing," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2499–2517, Apr. 2022, doi: 10.1109/TCOMM.2022.3151064.

[98] D. Zhang, Y. Ma, X. S. Hu, and D. Wang, "Toward privacy-aware task allocation in social sensing-based edge computing systems," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11384–11400, Dec. 2020, doi: 10.1109/JIOT.2020.2999025.

[99] C. Gamanayake, L. Jayasinghe, B. K. K. Ng, and C. Yuen, "Cluster pruning: An efficient filter pruning method for edge AI vision applications," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 4, pp. 802–816, May 2020, doi: 10.1109/JSTSP.2020.2971418.

[100] Y. Qian, J. Xu, S. Zhu, W. Xu, L. Fan, and G. K. Karagiannidis, "Learning to optimize resource assignment for task offloading in mobile edge computing," *IEEE Commun. Lett.*, early access, Mar. 16, 2022, doi: 10.1109/LCOMM.2022.3159742.

[101] A. Libri, A. Bartolini, and L. Benini, "PAElla: Edge AI-based real-time malware detection in data centers," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9589–9599, Oct. 2020, doi: 10.1109/JIOT.2020.2986702.

[102] T. Huang, T. Luo, M. Yan, J. Tianyi Zhou, and R. Goh, "RCT: Resource constrained training for edge AI," 2021, *arXiv:2103.14493*.
[103] P. Subedi, J. Hao, I. K. Kim, and L. Ramaswamy, "AI multi-tenancy on edge: Concurrent deep learning model executions and dynamic model placements on edge devices," in *Proc. IEEE 14th Int. Conf. Cloud Comput. (CLOUD)*, Chicago, IL, USA, Sep. 2021, pp. 31–42.
[104] P. Liu, J. Jiang, G. Zhu, L. Cheng, W. Jiang, W. Luo, Y. Du, and Z. Wang, "Training time minimization for federated edge learning with optimized gradient quantization and bandwidth allocation," 2021, *arXiv:2112.14387*.
[105] I. Chakraborty, D. Roy, I. Garg, A. Ankit, and K. Roy, "Constructing energy-efficient mixed-precision neural networks through principal component analysis for edge intelligence," *Nat. Mach. Intell.*, vol. 2, pp. 43–55, Jan. 2020, doi: 10.1038/s42256-019-0134-0.
[106] M. Gorsline, J. Smith, and C. Merkel, "On the adversarial robustness of quantized neural networks," in *Proc. Great Lakes Symp.*, New York, NY, USA, Jun. 2021, pp. 189–194.
[107] J. Liu, Y. Liu, and Q. Zhang, "A weight initialization method based on neural network with asymmetric activation function," *Neurocomputing*, vol. 483, pp. 171–182, Apr. 2022, doi: 10.1016/j.neucom.2022.01.088.
[108] Y. Chen, C. Hawkins, K. Zhang, Z. Zhang, and C. Hao, "3U-EdgeAI: Ultra-low memory training, ultra-low bitwidth quantization, and ultra-low latency acceleration," in *Proc. Great Lakes Symp.*, New York, NY, USA, Jun. 2021, pp. 157–162.
[109] B. Chen, A. Bakhshi, G. Batista, B. Ng, and T.-J. Chin, "Update compression for deep neural networks on the edge," 2022, *arXiv:2203.04516*.
[110] P. Andreev, A. Fritzler, and D. Vetrov, "Quantization of generative adversarial networks for efficient inference: A methodological study," 2021, *arXiv:2108.13996*.
[111] J. Shao and J. Zhang, "Communication-computation trade-off in resource-constrained edge inference," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 20–26, Dec. 2020, doi: 10.1109/MCOM.001.2000373.
[112] X. Yang, S. Hua, Y. Shi, H. Wang, J. Zhang, and K. B. Letaief, "Sparse optimization for green edge AI inference," *J. Commun. Inf. Netw.*, vol. 5, no. 1, pp. 1–15, Mar. 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9055106, doi: 10.23919/JCIN.2020.9055106.
[113] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep. 2019, doi: 10.1109/MNET.2019.1800286.
[114] K. Yang, Y. Zhou, Z. Yang, and Y. Shi, "Communication-efficient edge AI inference over wireless networks," 2020, *arXiv:2004.13351*.
[115] M. Li, J. Gao, C. Zhou, S. Xuemin, and W. Zhuang, "Slicing-based AI service provisioning on network edge," 2021, *arXiv:2105.07052*.
[116] L. Yang, Y. Lu, J. Cao, J. Huang, and M. Zhang, "E-tree learning: A novel decentralized model learning framework for edge AI," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11290–11304, Jul. 2021, doi: 10.1109/JIOT.2021.3052195.
[117] W. Wan, R. Kubendran, C. Schaefer, S. B. Eryilmaz, W. Zhang, D. Wu, S. Deiss, P. Raina, H. Qian, B. Gao, S. Joshi, H. Wu, H.-S. P. Wong, and G. Cauwenberghs, "Edge AI without compromise: Efficient, versatile and accurate neurocomputing in resistive random-access memory," 2021, *arXiv:2108.07879*.
[118] Y. Long, I. Chakraborty, G. Srinivasan, and K. Roy, "Complexity-aware adaptive training and inference for edge-cloud distributed AI systems," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Washington, DC, USA, Jul. 2021, pp. 573–583.
[119] Y. Du, S. Yang, and K. Huang, "High-dimensional stochastic gradient quantization for communication-efficient edge learning," *IEEE Trans. Signal Process.*, vol. 68, pp. 2128–2142, 2020, doi: 10.1109/TSP.2020.2983166.
[120] S. Khoram and J. Li, "TOCO: A framework for compressing neural network models based on tolerance analysis," 2019, *arXiv:1912.08792*.
[121] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, Jan. 2020, doi: 10.1109/TWC.2019.2946140.
[122] D. Liu, X. Chen, Z. Zhou, and Q. Ling, "HierTrain: Fast hierarchical edge AI learning with hybrid parallelism in mobile-edge-cloud computing," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 634–645, 2020, doi: 10.1109/OJCOMS.2020.2994737.
[123] S. Wang, Y. Hu, and J. Wu, "KubeEdge.AI: AI platform for edge devices," 2020, *arXiv:2007.09227*.
[124] H. Rexha and S. Lafond, "Data collection and utilization framework for edge AI applications," in *Proc. IEEE/ACM 1st Workshop AI Eng. Softw. Eng. AI (WAIN)*, Madrid, Spai, May 2021, pp. 105–108.
[125] D. Gerlinghoff, Z. Wang, X. Gu, R. S. M. Goh, and T. Luo, "E3NE: An end-to-end framework for accelerating spiking neural networks with emerging neural encoding on FPGAs," *IEEE Trans. Parallel Distrib. Syst.*, early access, Nov. 18, 2021, doi: 10.1109/TPDS.2021.3128945.
[126] Q. Liang, P. Shenoy, and D. Irwin, "AI on the edge: Rethinking AI-based IoT applications using specialized edge architectures," 2020, *arXiv:2003.12488*.
[127] C. Hao, Y. Chen, X. Zhang, Y. Li, J. Xiong, W.-M. Hwu, and D. Chen, "Effective algorithm-accelerator co-design for AI solutions on edge devices," in *Proc. Great Lakes Symp.*, New York, NY, USA, Sep. 2020, pp. 283–290.
[128] Q. Liang, W. A. Hanafy, A. Ali-Eldin, and P. Shenoy, "Model-driven cluster resource management for AI workloads in edge clouds," 2022, *arXiv:2201.07312*.
[129] H. Liao, Y. Mu, Z. Zhou, M. Sun, Z. Wang, and C. Pan, "Blockchain and learning-based secure and intelligent task offloading for vehicular fog computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4051–4063, Jul. 2021, doi: 10.1109/TITS.2020.3007770.
[130] M. Li, F. R. Yu, P. Si, R. Yang, Z. Wang, and Y. Zhang, "UAV-assisted data transmission in blockchain-enabled M2M communications with mobile edge computing," *IEEE Netw.*, vol. 34, no. 6, pp. 242–249, Nov. 2020, doi: 10.1109/MNET.011.2000147.
[131] M. Liu, Y. Teng, F. R. Yu, V. C. M. Leung, and M. Song, "A mobile edge computing (MEC)-enabled transcoding framework for blockchain-based video streaming," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 81–87, Apr. 2020, doi: 10.1109/MWC.001.1800332.
[132] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4321–4334, Jun. 2020, doi: 10.1109/TWC.2020.2982627.
[133] C.-H. Chu, "Task offloading based on deep learning for blockchain in mobile edge computing," *Wireless Netw.*, vol. 27, no. 1, pp. 117–127, Jan. 2021, doi: 10.1007/s11276-020-02444-7.
[134] F. Guo, F. R. Yu, H. Zhang, H. Ji, M. Liu, and V. C. M. Leung, "Adaptive resource allocation in future wireless networks with blockchain and mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1689–1703, Mar. 2020, doi: 10.1109/TWC.2019.2956519.
[135] Y. Dai, D. Xu, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4312–4324, Apr. 2020, doi: 10.1109/TVT.2020.2973705.
[136] P. Zhang, X. Pang, N. Kumar, G. S. Aujla, and H. Cao, "A reliable data-transmission mechanism using blockchain in edge computing scenarios," *IEEE Internet Things J.*, early access, Sep. 3, 2020, doi: 10.1109/JIOT.2020.3021457.
[137] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan, and Y. Zhang, "Blockchain empowered cooperative authentication with data traceability in vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4221–4232, Apr. 2020, doi: 10.1109/TVT.2020.2969722.
[138] S. Guo, Y. Dai, S. Guo, X. Qiu, and F. Qi, "Blockchain meets edge computing: Stackelberg game and double auction based task offloading for mobile blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5549–5561, May 2020, doi: 10.1109/TVT.2020.2982000.
[139] Y. Gao, W. Wu, P. Si, Z. Yang, and F. R. Yu, "B-ReST: Blockchain-enabled resource sharing and transactions in fog computing," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 172–180, Apr. 2021, doi: 10.1109/MWC.001.2000102.
[140] R. Gupta, D. Reebadiya, S. Tanwar, N. Kumar, and M. Guizani, "When blockchain meets edge intelligence: Trusted and security solutions for consumers," *IEEE Netw.*, vol. 35, no. 5, pp. 272–278, Sep. 2021, doi: 10.1109/MNET.001.2000735.
[141] H. Zhang, R. Wang, W. Sun, and H. Zhao, "Mobility management for blockchain-based ultra-dense edge computing: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7346–7359, Nov. 2021, doi: 10.1109/TWC.2021.3082986.
[142] A. V. Rivera, A. Refaey, and E. Hossain, "A blockchain framework for secure task sharing in multi-access edge computing," *IEEE Netw.*, vol. 35, no. 3, pp. 176–183, May 2021, doi: 10.1109/MNET.011.2000497.

[143] S. Islam, S. Badsha, S. Sengupta, H. La, I. Khalil, and M. Atiquzzaman, "Blockchain-enabled intelligent vehicular edge computing," *IEEE Netw.*, vol. 35, no. 3, pp. 125–131, May 2021, doi: 10.1109/MNET.011.2000554.
[144] A. Gumaei, M. Al-Rakhami, M. M. Hassan, P. Pace, G. Alai, K. Lin, and G. Fortino, "Deep learning and blockchain with edge computing for 5G-enabled drone identification and flight mode detection," *IEEE Netw.*, vol. 35, no. 1, pp. 94–100, Jan. 2021, doi: 10.1109/MNET.011.2000204.
[145] G. Li, X. Ren, J. Wu, W. Ji, H. Yu, J. Cao, and R. Wang, "Blockchain-based mobile edge computing system," *Inf. Sci.*, vol. 561, pp. 70–80, Jun. 2021, doi: 10.1016/j.ins.2021.01.050.
[146] J. Zhang, L. Yuan, and S. Xu, "A lightweight blockchain-based access control scheme for integrated edge computing in the Internet of Things," 2021, *arXiv:2111.06544*.
[147] D. C. Nguyen, S. Hosseinalipour, D. J. Love, P. N. Pathirana, and C. G. Brinton, "Latency optimization for blockchain-empowered federated learning in multi-server edge computing," 2022, *arXiv:2203.09670*.
[148] Z. Wang, "Development status and trend prospect of edge computing," *Auto. Expo*, vol. 38, no. 2, pp. 22–29, 2021.
[149] D. Zeng, L. Cheng, L. Gu, and Y. Li, "Cloud native based edge computing: Vision and challenges," *Chin. J. Int. Things*, vol. 5, pp. 7–17, Jan. 2021, doi: 10.11959/j.issn.2096-3750.2021.00206.
[150] Y. Xiong, Y. Sun, L. Xing, and Y. Huang, "Extend cloud to edge with KubeEdge," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Seattle, WA, USA, Oct. 2018, pp. 373–377.
[151] J. John, A. Ghosal, T. Margaria, and D. Pesch, "DSLs for model driven development of secure interoperable automation systems with EdgeX foundry," in *Proc. Forum Specification Design Lang. (FDL)*, Antibes, France, Sep. 2021, pp. 1–8.
[152] Z. Qin, H. Wang, Y. Qu, H. Dai, and Z. Wei, "Air-ground collaborative mobile edge computing: Architecture, challenges, and opportunities," 2021, *arXiv:2101.07930*.

**BIN LIU** received the B.S. degree in electrical engineering and its automation from the Sichuan University of Science and Engineering, Yibin, China, in 2020, where he is currently pursuing the M.S. degree. His main research interest includes distributed processing.

**ZHONGQIANG LUO** (Member, IEEE) received the B.S. and M.S. degrees in communication engineering and pattern recognition and intelligent systems from the Sichuan University of Science and Engineering, China, in 2009 and 2012, respectively, and the Ph.D. degree in communication and information systems from the University of Electronic Science and Technology of China (UESTC), in 2016. Since 2017, he has been with the Sichuan University of Science and Engineering, where he is currently an Associate Professor. From December 2018 to December 2019, he was a Visiting Scholar with the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County (UMBC). His research interests include information fusion, blind source separation, signal processing for wireless communication systems, and intelligent signal processing.

**HONGBO CHEN** received the M.S. degree in electrical engineering and automation from Wuhan University, Wuhan, China, in 2012. As a main researcher, he participated in "Research and Application of Non-Electric Quantity Testing Technology in Condition Based Maintenance of Power Transmission and Transformation Equipment," "Detection and Evaluation of Insulation Defects and Pollution Ultraviolet Rays of High Voltage Power Transmission and Transformation Equipment," "Pollution Investigation, Measurement and Analysis Research along the Sichuan Section of Xiangjiaba-Shanghai and Jinping-Sunan ±800 kV UHV DC Transmission Lines," and other key scientific and technological projects of State Grid Corporation of China.

**CHENGJIE LI** received the B.Sc. degree from Shandong Normal University, Qufu, China, in 2004, the M.Sc. degree in computer software and theory from Xihua University, Chengdu, China, in 2009, and the Ph.D. degree in communication and information system from the University of Electronic Science and Technology of China (UESTC), Chengdu, in 2017. Since 2017, he has been with Southwest Minzu University, where he is currently a Lecturer. His research interests include blind source separation, data mining, and intelligent information processing.